



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Daniela Backhaus

Untersuchung des hydrodynamischen Einflusses eines Haltearms auf ein Freifahrtgerät anhand von CFD

*Fakultät Technik und Informatik
Department Maschinenbau und Produktion*

*Faculty of Engineering and Computer Science
Department of Mechanical Engineering and
Production Management*

Daniela Backhaus

**Untersuchung des hydrodynamischen
Einflusses eines Haltearms auf ein
Freifahrtgerät anhand von CFD**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Maschinenbau/Energie- und Anlagensysteme
am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

in Zusammenarbeit mit:
Hamburgische Schiffbau-Versuchsanstalt GmbH
Konstruktion und Seakeeping, Manoeuvring and Offshore
Bramfelder Straße 164
22305 Hamburg

Erstprüfer: Prof. Dr.-Ing. Peter Wulf
Zweitprüfer: Dipl.-Ing. Jan Lassen

Abgabedatum: 24.05.2018

Zusammenfassung

Daniela Backhaus

Thema der Bachelorthesis

Untersuchung des hydrodynamischen Einflusses eines Haltearms auf ein Freifahrtgerät anhand von CFD

Stichworte

CFD, numerische Strömungsberechnung, Freifahrtversuch, Düse

Kurzzusammenfassung

In dieser Arbeit wurden mittels des CFD-Programms OpenFOAM numerische Strömungssimulationen durchgeführt. Ziel der Simulation war es, den Einfluss eines Haltearms, der eine Düse an einem Freifahrtgerät anbringt, zu untersuchen. Hierzu wurde mit dem Programm snappyHexMesh ein numerisches Gitter erstellt, welches verschiedene Verfeinerungen und die zu untersuchende Geometrie beinhaltet. Die Simulation wurde mit unterschiedlichen Gitterauflösungen durchgeführt. Anschließend wurden die Ergebnisse mit dem Programm ParaView visualisiert und danach interpretiert.

Daniela Backhaus

Title of the paper

Study of the hydrodynamic influence of a holder on an Open Water Test device based on CFD

Keywords

CFD, numerical fluid dynamics, open water test, duct, nozzle

Abstract

In this thesis numerical flow simulations were performed using the CFD software OpenFOAM. The goal of the simulation was the investigation of the hydrodynamic influence of a holder, which attaches a nozzle on an open water test device. To accomplish this, a numerical mesh has been created, using the software snappyHexMesh. The mesh consists of various refinements, as well as the geometry, which is to be examined. The simulation was carried out with different resolutions of the mesh. Subsequently the results were visualised using the software ParaView and interpreted afterwards.

Inhaltsverzeichnis

Aufgabenstellung	VII
Abbildungsverzeichnis	VII
Tabellenverzeichnis	X
Symbolverzeichnis	XIII
Abkürzungsverzeichnis	XVII
1 Einleitung	1
2 Hamburgische Schiffbau-Versuchsanstalt	3
3 Freifahrtgerät	7
3.1 Der Freifahrtversuch	7
3.2 Problemstellung	9
3.3 Freifahrtgerät H39	10
4 Software	13
4.1 OpenFOAM	13
4.2 Vernetzer snappyHexMesh	15
4.3 Post-Processing mit ParaView/paraFoam	15
5 Diskretisierung und Numerik	17
5.1 Theoretische Grundlagen zur Strömungsberechnung	17
5.2 Netz-Generierung	21
5.3 Randbedingungen	21
5.4 Turbulenzmodell	22
5.5 Wandfunktionen	22
5.6 Approximationen	23
5.7 Lösungsalgorithmen	24
6 Simulation des Freifahrtgeräts	27
6.1 Gitteraufbau	27
6.2 Randbedingungen	40
6.3 Simulationseinstellungen	41
6.4 Implementierung eines Propellers	43

7 Vergleich der Ergebnisse	47
7.1 Geschwindigkeitsverläufe	48
7.2 Druckverteilung	62
7.3 Kräfte	68
7.4 Interpolierte Darstellungen	70
8 Fazit	75
9 Ausblick	77
10 Zusammenfassung	79
Literatur	80
A Software-Einstellungen	83
A.1 system	83
A.2 constant	97
A.3 0.Orig	99
A.4 Skripte	108
B Zusätzliche Abbildungen	111
B.1 Abbildungen zu Kapitel 6	111
B.2 Abbildungen zu Kapitel 7	112
C Weitere Informationen	123
C.1 Dateien für die actuatorDisk	123

Aufgabenstellung

Die Hamburgische Schiffbau-Versuchsanstalt GmbH (HSVA) führt unter anderem Messungen an Propellern mittels Freifahrtversuchen durch. Das derzeit genutzte Freifahrtmessgerät soll um die Möglichkeit, auch Propeller mit Düse messen zu können, erweitert werden. Dafür wird ein zusätzlicher Haltearm senkrecht über der Düse angebracht, in dem eine Kraftmessdose verbaut ist.

Der Haltearm ist nach Erfahrungswerten strömungsgünstig geformt, stellt aber naturgemäß eine Störung in der Strömung dar.

In der Bachelorthesis soll untersucht werden, ob bzw. welchen hydrodynamischen Einfluss der Haltearm auf die Kräfte, genauer gesagt die Kraftmessungen, hat.

Diese Untersuchung soll mittels CFD (Computational Fluid Dynamics) erfolgen.

Im Einzelnen sind für die Bachelorthesis folgende Aufgaben vorgesehen:

- Simulation des neu konstruierten Haltearms inkl. Düse
- Auswahl eines geeigneten Solvers
- Erstellung eines geeigneten Netzes
- Düse und Propeller werden stationär modelliert
- Propeller wird nicht drehend modelliert
- Post-Processing
- Dokumentation der Ergebnisse

Abbildungsverzeichnis

2.1	Kavitationstank HYKAT [8]	4
2.2	Großer Eistank [8]	5
2.3	Großer Eistank Schema [8]	5
2.4	Großer Schlepptank [8]	6
2.5	Großer Schlepptank Schema [8]	6
3.1	Beispielhaftes Freifahrtprogramm [12]	9
3.2	Schema des Freifahrtgeräts mit Düse [7]	10
3.3	Freifahrtgerät H32 mit Düse frontal [7]	11
3.4	Freifahrtgerät H32 mit Düse [7]	11
4.1	Case-Struktur in OpenFOAM [5]	14
4.2	Schema der Vernetzung mit snappyHexMesh (in Anlehnung an[5])	15
5.1	Geschwindigkeitsprofile in der viskosen Unterschicht und der wandnahen Schicht [13]	23
5.2	Schema zur Berechnung des Druckes und Erfüllung der Kontinuitätsgleichung nach dem SIMPLE-Verfahren [13]	25
6.1	blockMesh Ansicht	28
6.2	Netzverfeinerung an einer Blockkante [5]	29
6.3	blockMesh grading calculation [6]	29
6.4	blockMesh Gesamtansicht	30
6.5	blockMesh Detail	31
6.6	Ausschnitt des Netzes an der Geometrie	32
6.7	Ausschnitt des Netzes an der Halterung	33
6.8	Ausschnitt des Netzes an Düse unten	33
6.9	Ausschnitt des Netzes an der vorderen Kante	34
6.10	Ausschnitt des Netzes an der Hinterkante	34
6.11	y^+ -Verteilung	35
6.12	y^+ -Verteilung im Inneren der Düse	36
6.13	Residuenverlauf Gitter mit 820k Zellen	37
6.14	Residuenverlauf Gitter mit 2.1M Zellen	37
6.15	Residuenverlauf Gitter mit 5.6M Zellen	38
6.16	Kräfte zur Gitterunabhängigkeitsanalyse	38
6.17	Kraftverlauf der Gitter mit 820k Zellen, 2.1M Zellen, 5.6M Zellen - mit Haltearm	39
6.18	Kraftverlauf der Gitter mit 653k Zellen, 1.7M Zellen, 4.6M Zellen - nur Düse	39
6.19	Schema Berechnungsgebiet	40

6.20	Diagramm Schubbeiwert c_T über den axialen Induktionsfaktor a [3]	44
7.1	Geschwindigkeit in x-Richtung - Gesamtansicht - mit Halter	49
7.2	Geschwindigkeit in x-Richtung - Gesamtansicht - ohne Halter	49
7.3	Geschwindigkeit in x-Richtung - Detailansicht - mit Halter	51
7.4	Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter	51
7.5	Geschwindigkeit in x-Richtung - Stromlinien perspektivisch - mit Halter	53
7.6	Geschwindigkeit in x-Richtung - Stromlinien perspektivisch - ohne Halter	53
7.7	Geschwindigkeit in x-Richtung - Vektorpfeile - mit Halter	55
7.8	Geschwindigkeit in x-Richtung - Vektorpfeile - ohne Halter	55
7.9	Stromlinien im Raum - mit Halter	57
7.10	Stromlinien im Raum - ohne Halter	57
7.11	Stromlinien in der xy-Ebene - mit Halter	59
7.12	Stromlinien in der xy-Ebene - ohne Halter	59
7.13	Stromlinien um den Haltearm	61
7.14	Druck - mit Halter	63
7.15	Druck - ohne Halter	63
7.16	Druck auf Oberfläche Außenseite - mit Halter	65
7.17	Druck auf Oberfläche Außenseite - ohne Halter	65
7.18	Druck auf Oberfläche Innenseite - mit Halter	67
7.19	Druck auf Oberfläche Innenseite - ohne Halter	67
7.20	Kräfte auf die Düse über Iterationen	69
7.21	Strömung in der xz-Ebene - mit Halter	71
7.22	Strömung in der xz-Ebene - ohne Halter	71
7.23	Strömung um die Düse in der yz-Ebene - mit Halter	73
7.24	Strömung um die Düse in der yz-Ebene - ohne Halter	73
B.1	Geometrie des Haltearms mit Düse	111
B.2	Geschwindigkeit in x-Richtung - Detailansicht - mit Halter - Iteration 700	112
B.3	Geschwindigkeit in x-Richtung - Detailansicht - mit Halter - Iteration 800	113
B.4	Geschwindigkeit in x-Richtung - Detailansicht - mit Halter - Iteration 900	114
B.5	Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter - Iteration 700	115
B.6	Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter - Iteration 800	116
B.7	Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter - Iteration 900	117
B.8	Druck auf Oberfläche Außenseite - mit Halter - Iteration 800	118
B.9	Druck auf Oberfläche Außenseite - mit Halter - Iteration 900	119
B.10	Druck auf Oberfläche Außenseite - ohne Halter - Iteration 800	119
B.11	Druck auf Oberfläche Außenseite - ohne Halter - Iteration 900	120
B.12	Druck auf Oberfläche Innenseite - mit Halter - Iteration 800	120
B.13	Druck auf Oberfläche Innenseite - mit Halter - Iteration 900	121
B.14	Druck auf Oberfläche Innenseite - ohne Halter - Iteration 800	122
B.15	Druck auf Oberfläche Innenseite - ohne Halter - Iteration 900	122

Tabellenverzeichnis

6.1	simpleGrading blockMeshDict	27
6.2	Abmessungen blockMesh	28
6.3	Randbedingungen für U, p und ν_t	40
6.4	Randbedingungen für k und ω	41
6.5	Solver Einstellungen aus <i>fvSolution</i>	42
7.1	Mittelwerte der Kraft auf die Düse	68

Symbolverzeichnis

Lateinische Symbole

Symbol	Einheit	Beschreibung
A	m^2	Fläche
a, b	–	Variable für System der Finiten Volumen Methode
a	–	axialer Induktionsfaktor
c_D	–	Widerstandskoeffizient
C_μ	–	Konstante
c_P	–	Leistungskoeffizient
c_T	–	Schubkoeffizient
D	m	Durchmesser
D_{ij}	–	Tensor der Deformationsrate
F	N	Kraft
f	–	generisches Formelzeichen
g	m/s^2	Erdbeschleunigung
J	–	Fortschrittsgrad
K_Q	–	Drehmomentenkoeffizient
K_T	–	Schubkoeffizient
\mathbf{k}	$\text{kg}/(\text{s}^2 \text{m}^2)$	Volumenkraftvektor
k	J/kg	Kinetische Energie der Turbulenz
l	m	Länge
L_T	m	turbulentes Längenmaß

Lateinische Symbole Fortsetzung

Symbol	Einheit	Beschreibung
m	kg	Masse
n	1/s	Drehzahl
\mathbf{n}	–	Einheitsvektor
p	N/m ²	Druck
Q	Nm	Drehmoment
R	m	Radius
Re	–	Reynoldszahl
r_k	–	Zellverfeinerungsfaktor
S	m ²	Oberfläche
t	s	Zeit
T	N	Schub
T_{ij}	N/m ²	Spannungstensor
Tu	–	Turbulenzintensität
U	m/s	Geschwindigkeit
\mathbf{v}	m/s	Geschwindigkeitsvektor
v_a	m/s	Geschwindigkeit des Schleppwagens
V	m ³	Volumen
x, y, z	–	Koordinaten
y	mm	Wandabstand
y^+	–	dimensionslose Zellhöhe

Griechische Symbole

Symbol	Einheit	Beschreibung
Γ	m^2/s	Diffusionskoeffizient
$\delta_{e,s}$	mm	Zelllänge Grading
$\delta_{i,j}$	–	Koenecker Symbol
ε	$\text{J}/(\text{kg s})$	Dissipationsrate
η_0	–	Freifahrtwirkungsgrad
μ	Pa s	dynamische Viskosität
μ_t	Pa s	Turbulenzviskosität
ν	m^2/s	kinematische Viskosität
ν_t	m^2/s	turbulente Scheinviskosität
ρ	kg/m^3	Dichte
τ	N/m^2	Schubspannung
ϕ	–	generische Größe
ω	1/s	charakteristische Frequenz der energiedissipierenden Wirbel

Notationen

f'	Abweichung vom Mittelwert
$(\dots)'$	Korrekturwert
\bar{f}	Mittelwert
$(\dots)_{NB}$	Nachbarzelle
$(\dots)^*$	Schätzwert
\mathbf{f}	Vektor
$(\dots)^m$	Zähler der äußeren Iteration
$(\dots)_P$	Zell- oder Flächenmittelpunkt

Anmerkung: Bei der Zellanzahl bedeutet k Tausend und M Millionen. $820k$ entsprechen 820.000 Zellen.

Abkürzungsverzeichnis

CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CPMC	Computerized Planar Motion Carriage
DGL	Differentialgleichung
DNS	Direkte numerische Simulation
FDM	Finite Differenzen Methode
FEM	Finite Elemente Methode
FVM	Finite Volumen Methode
GAMG	Geometric Agglomerated Algebraic Multigrid Solver
GUI	Graphical User Interface
HSVA	Hamburgische Schiffbau-Versuchsanstalt
ITTC	International Towing Tank Conference
KM	Kontrollmasse
KV	Kontrollvolumen
LES	Large Eddy Simulation
LIC	Line Integral Convolution
MW	Mittelwert
OBJ	Wavefront Object Dateiformat
OpenFOAM	Open Source Field Operation and Manipulation
RANS	Reynolds-Averaged-Navier-Stokes
SIMPLE	Semi-implicit Method for Pressure Linked Equations
SST	Shear Stress Transport
STL	Stereolithography Dateiformat

1 Einleitung

In der 1913 gegründeten Hamburgischen Schiffbau-Versuchsanstalt (HSVA) werden diverse schiffstechnische Versuche durchgeführt. Zur Bestimmung der technisch und wirtschaftlich wichtigen Propeller-Leistung werden unter anderem Propulsions- und Freifahrtversuche in dem 300 m langen Schlepptank durchgeführt. Um die benötigte Leistung zum Antrieb eines Schiffes bestimmen zu können, werden beide Versuche benötigt. In dieser Arbeit soll das für die Freifahrtversuche benötigte Freifahrtgerät geprüft werden. Um auch Propeller beurteilen zu können, die mit einer Düse zusammen betrieben werden, soll das Freifahrtgerät H39 mit einem Haltearm ausgestattet werden, an dem solch eine Düse befestigt und gemessen werden kann. Es soll mithilfe einer CFD(Computational Fluid Dynamics)-Simulation geprüft werden, ob und welchen hydrodynamischen Einfluss dieser Halterarm auf die Messergebnisse der Kraftmessung hat. Zu diesem Zweck soll mit dem CFD-Programm OpenFOAM ein Simulationsfall erstellt werden. Die Vorgehensweise in OpenFOAM entspricht der allgemeinen im Rahmen von CFD-Simulationen. Die erste Phase beinhaltet das Pre-Processing, in dem ein numerisches Gitter erzeugt wird und die immens wichtigen Randbedingungen festgelegt werden. Die zweite Phase ist die, in der der Anwender wenig beisteuert und der Computer die numerischen Arbeiten übernimmt, indem er Gleichungssysteme löst. In der dritten Phase, dem Post-Processing, werden die Ergebnisse überprüft und visualisiert. Mit dem mitgelieferten Utility *snappyHexMesh* wird zu diesem Zweck ein numerisches Gitter aufgebaut und die Rechnung erfolgt mit einem in OpenFOAM implementierten Gleichungslöser. Die Ergebnisse werden mit der Software ParaView visualisiert und anschließend interpretiert und diskutiert. Der Vergleich der Kräfte, die auf die Düse wirken, soll Aufschluss darüber geben, ob sich ein hydrodynamischer Einfluss des Haltearms ausbildet.

2 Hamburgische Schiffbau-Versuchsanstalt

Die Hamburgische Schiffbau-Versuchsanstalt GmbH (kurz HSVA) ist eine private Non-Profit-Organisation, die Forschung und Beratung für die weltweite maritime Industrie bietet.

Gegründet wurde die HSVA 1913. Heute bietet sie Raum für Versuche für „alles, was fließt“.

Die Hauptthemengebiete sind in folgende Abteilungen unterteilt:

- Resistance and Propulsion
- Propellers and Cavitation
- Arctic Technology
- Computational Fluid Dynamics
- Seakeeping, Manoeuvring and Offshore
- Administration, Workshops and Tank Operation

Die meisten Versuche finden in den folgenden Einrichtungen statt:

HYKAT

Der HYKAT ist ein Umlauftank, in dem Kavitation untersucht werden kann. Hierfür stehen Hochgeschwindigkeits-Videosysteme zur Verfügung. Die maximale Geschwindigkeit in der Teststrecke beläuft sich auf 12,6 m/s. Der Testbereich hat die Abmessungen 2,8 m × 1,6 m × 11 m und bietet somit Platz für ganze Schiffsmodelle [8].

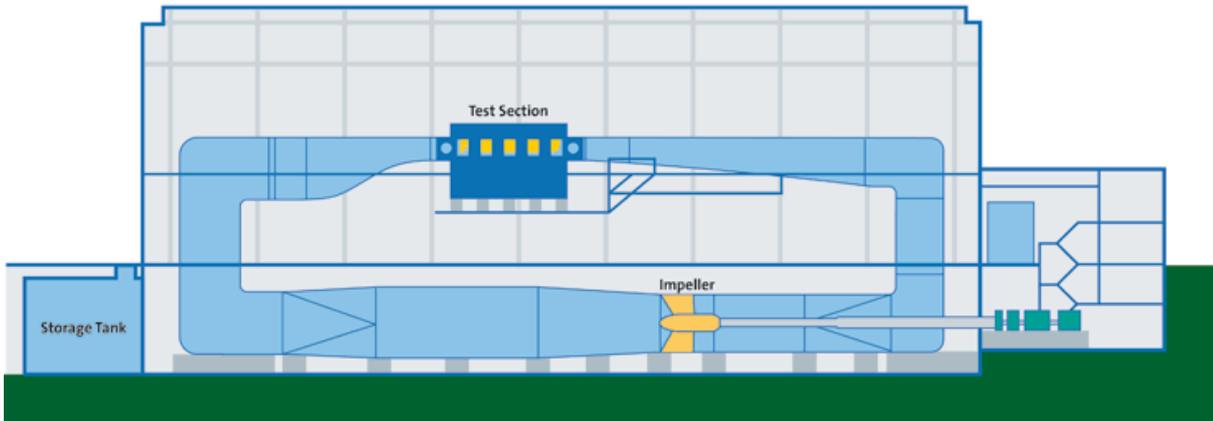


Abbildung 2.1: Kavitationstank HYKAT [8]

Eistank

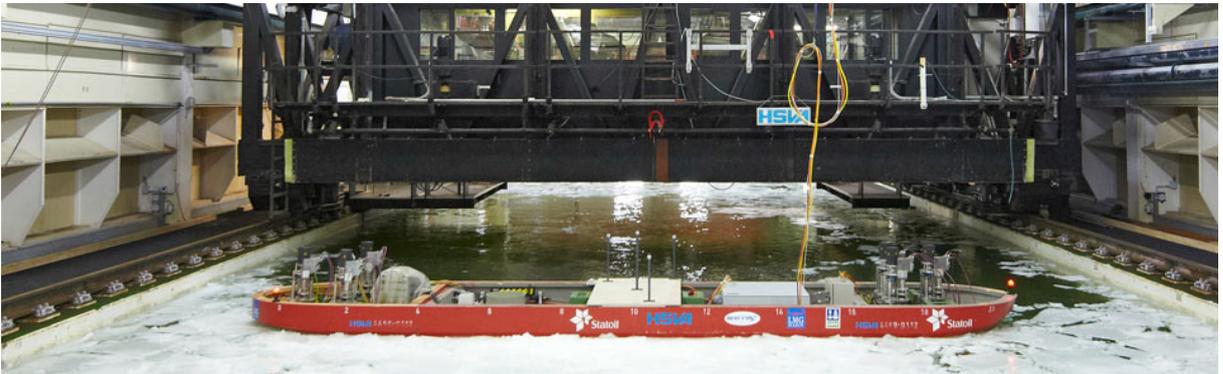


Abbildung 2.2: Großer Eistank [8]

Der große Eistank der HSVA ist 78 m lang, 10 m breit und 2,5 m tief. Am Ende des Tanks befindet sich ein Tiefwasserbereich mit den Abmessungen ($L \times B \times T$) 12 m \times 10 m \times 5 m. Ein Belüftungssystem kühlt die Raumluft im Eistank auf bis zu -20°C herunter und lässt das Salzwasser mit einer Rate von 2 mm/h gefrieren. Der Schleppwagen erreicht 3 m/s und hat eine Schleppkraft von bis zu 50 kN [8].

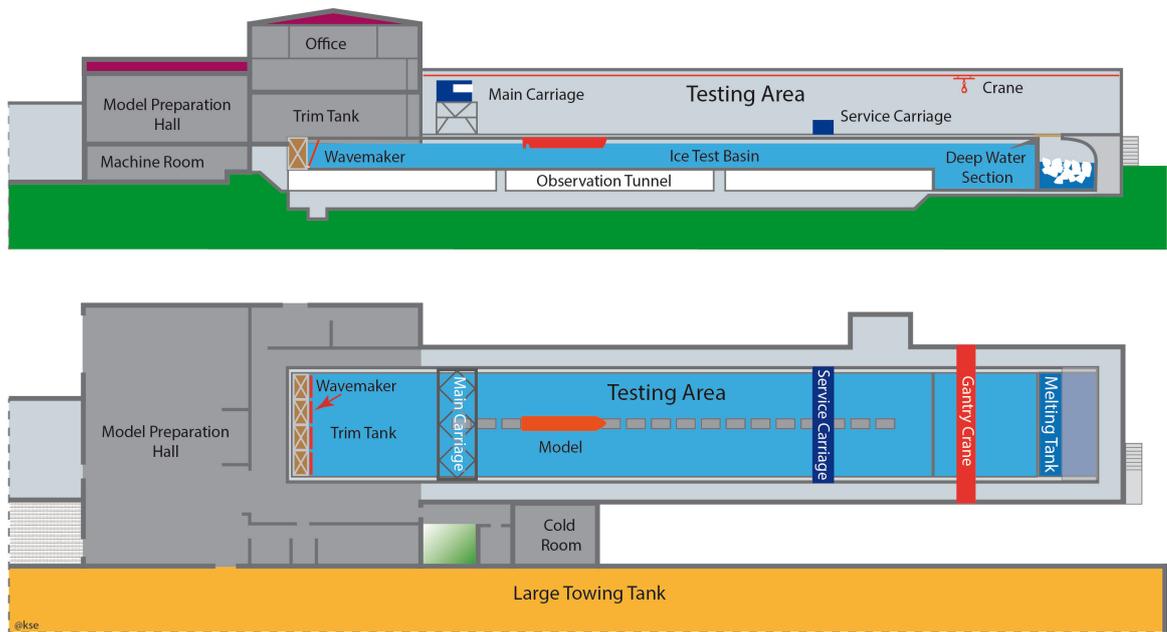


Abbildung 2.3: Großer Eistank Schema [8]

Großer Schlepptank



Abbildung 2.4: Großer Schlepptank [8]

Der große Schlepptank (Large Towing Tank) ist ($L \times B \times T$) 300 m \times 18 m \times 6 m groß und wird bemannt betrieben. Die Modelle können bis zu 12 m groß sein. Am Ende und an den Seiten des Tanks befinden sich Wellenerzeuger, die periodische und aperiodische Seegangssituationen abbilden können. Hier werden unter anderem Widerstands-, Propulsions-, Freifahrt- oder auch Seegangsversuche durchgeführt. Bei diesen Versuchen kann das Modell geschleppt werden oder aber auch durch den „Computerized Planar Motion Carriage“ (CPMC) vorgegebenen Bewegungen folgen.

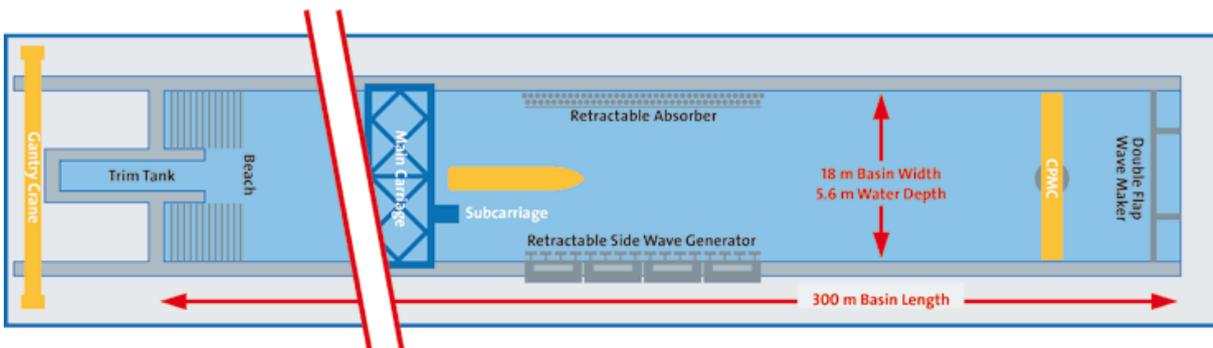


Abbildung 2.5: Großer Schlepptank Schema [8]

Neben den vorgestellten Versuchseinrichtungen verfügt die HSVA noch über je einen kleineren Schlepp- und Eistank, einen mittelgroßen Kavitationstank und einen großen Hochgeschwindigkeitskavitationstank. Auch verfügt die HSVA über einen Computercluster, der mit dem von der TU Hamburg-Harburg und der HSVA entwickelten CFD Code *FreSCo*⁺ betrieben wird. Außerdem werden die Modelle in den HSVA-eigenen Modellwerkstätten (Workshops) teils automatisiert (z.B. mit einer Fräse) und teils in Handarbeit gefertigt.

3 Freifahrtgerät

In der Hamburgischen Schiffbau-Versuchsanstalt werden verschiedenste Versuche zur Untersuchung von Schiffen und deren Bauteilen durchgeführt. Ein wichtiger Parameter in der Untersuchung und Auslegung von Schiffen ist die Leistung. Die Fragestellung, wie viel Leistung ein Schiffsantrieb benötigt, ist in konstruktiver und auch wirtschaftlicher Hinsicht von großer Bedeutung. Diese Fragestellung soll in Modellversuchen beantwortet werden. In Propulsionsversuchen wird ein Schiffsmodell mit eingebautem Propeller durch den Schlepptank gezogen. Da das Modell jedoch viel kleiner als das echte Schiff ist, aber das Fluid und damit die Viskosität in beiden Fällen Wasser ist, verhält sich die Strömung im Modellversuch anders als in der Großausführung. Die Reynoldszahlen bewegen sich im Modellversuch in einer Größenordnung von $Re_{\text{Modell}} \approx 10^6$ und in der Großausführung $Re_{\text{Großausführung}} \approx 10^9$. Die Grenzschicht am Modell ist wesentlich dicker, da sich das Wasser im Vergleich zur Großausführung viel zäher verhält. Dadurch weist der Nachstrom des Schiffsmodells einen anderen Verlauf als in der Großausführung auf und somit wird der Propeller anders angeströmt. Außerdem führt die geringere Reynoldszahl zu einem überproportional erhöhten Reibungsanteil. Der Modellpropeller wird dadurch höher belastet, als bei einem Schiff in Originalgröße. Um diesen Effekt zu vermindern, wird eine zusätzliche Kraft, als Reibungsabzug bezeichnet, in Fahrtrichtung aufgebracht. Der Nachstrom bleibt davon weitestgehend unberührt. Um also den Propeller und dessen Leistung unabhängig beurteilen zu können, wird der Freifahrtversuch durchgeführt, bei dem der Propeller homogen angeströmt wird.

3.1 Der Freifahrtversuch

Bei einem Freifahrtversuch wird ein Modellpropeller auf ein Freifahrtgerät montiert und bei konstanter Drehzahl n und einer variablen Geschwindigkeit v_a mit dem Schleppwagen durch das Wasser gezogen. Dabei wird der Schub T und das Drehmoment Q gemessen. Wie eingangs beschrieben verhält sich jedoch das Wasser im Modellversuch anders als im Vergleich zur Großausführung. Ein gravierender Unterschied ist, dass die Geschwindigkeit, die vom Schleppwagen vorgegeben wird, nicht die gleiche Geschwindigkeit ist, mit der ein Propeller hinter einem Schiff angeströmt wird, da der Nachstrom verzögert ist. Auch die Reibung an den Propellerflügeln ist durch die andere Grenzschichtausbildung ganz anders. Aus den gegebenen Werten wird der Fortschrittsgrad J

$$J = \frac{v_a}{nD} \quad (3.1)$$

mit dem Durchmesser D des Propellers berechnet. Hiermit lassen sich der Schubkoeffizient K_T

$$K_T = \frac{T}{\rho n^2 D^4} \quad (3.2)$$

und Drehmomentenkoeffizient K_Q

$$K_Q = \frac{Q}{\rho n^2 D^5} \quad (3.3)$$

berechnen.

Aus genannten Zusammenhängen lässt sich der Freifahrtwirkungsgrad η_0

$$\eta_0 = \frac{T v_a}{2\pi n Q} = \frac{K_T J}{K_Q 2\pi} \quad (3.4)$$

bestimmen.

Die Werte K_T und $10K_Q$ (der Faktor 10 dient der besseren Darstellbarkeit) werden über den Fortschrittsgrad J aufgetragen. Wichtige Eckpunkte im Diagramm sind

- der Punkt, bei welchem Fortschrittsgrad J der Freifahrtwirkungsgrad maximal wird
- Pfahlzug (auch „bollard pull“ genannt), bei dem der Fortschrittsgrad gleich null ist
- der Nullschubpunkt. Dieser stellt dar, dass, obwohl kein Schub generiert wird, ein Drehmoment aufgebracht werden muss. Dies ist der Reibung geschuldet, die überwunden werden muss, um den Propeller zu drehen
- der Punkt, an dem der Wirkungsgrad η_0 maximal wird

In Abbildung 3.1 erkennt man diese Punkte. Hier wird auch deutlich, dass die Umrechnung auf die Großausführung von immenser Bedeutung ist. Die Werte sind über den Fortschrittsgrad aufgetragen. Dieser ist nach Gleichung 3.1 von der Geschwindigkeit und der Drehzahl abhängig. Sollte hier also keine angemessene Umrechnung stattfinden und der tatsächliche Fortschrittsgrad wäre in der Großausführung höher, könnte der Wirkungsgrad vom Maximum zu sehr viel schlechteren Werten nach rechts abfallen.

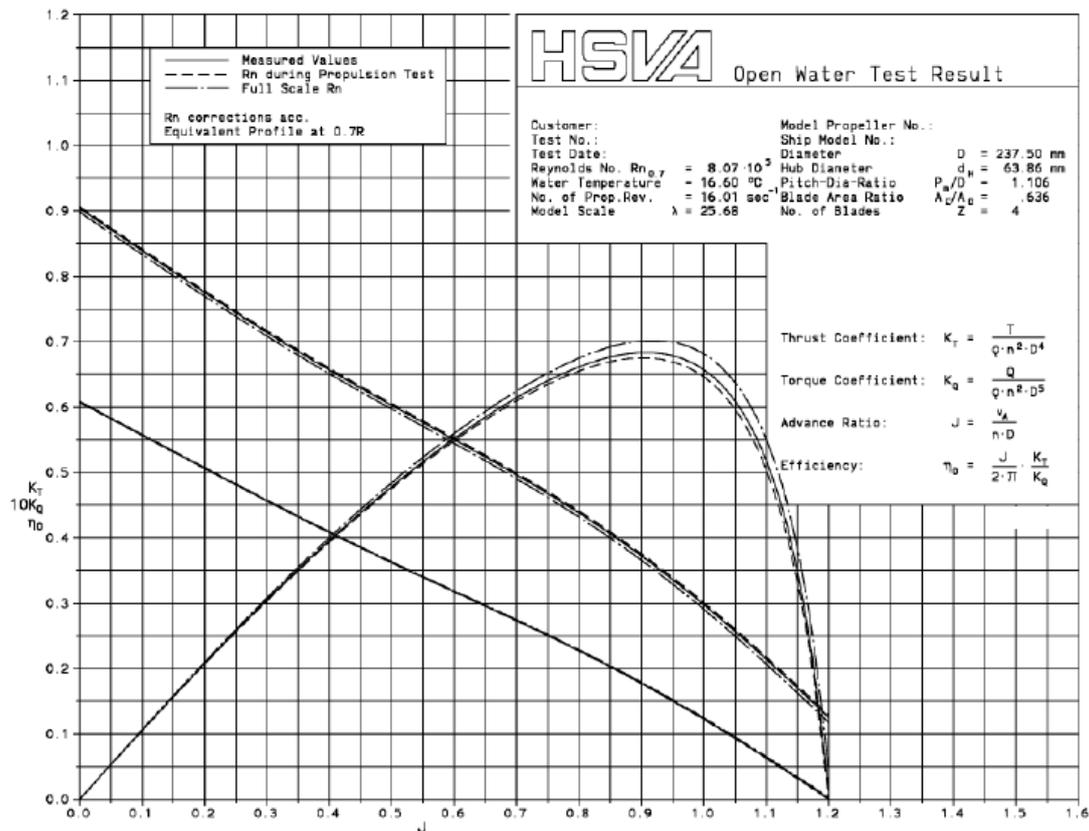


Abbildung 3.1: Beispielhaftes Freifahrtendiagramm [12]

Um diese Effekte zu berücksichtigen, wird in der HSVA die Streifenmethode nach Streckwall [7] verwendet. Hierzu werden die Propeller rechnerisch in typischerweise 10 Streifen geteilt und jeder Streifen hat eine lokale Reynoldszahl, basierend auf der individuellen Umfangsgeschwindigkeit und Blattlänge. Für jeden Streifen wird so der Widerstandskoeffizient c_D berechnet. Dies wird sowohl für die Reynoldszahlen des Modells, als auch für die Großausführung durchgeführt. Die Abweichung der Widerstandskoeffizienten Δc_D wird danach in eine Axial- und eine Umfangskomponente, je nach Neigung des Propellerblattes, aufgeteilt. Zum Schluss werden alle Werte aufsummiert, um die ΔK_Q - und ΔK_T -Werte für den gesamten Propeller zu bestimmen.

Für weiterführende Informationen siehe [2], [11] und [12].

3.2 Problemstellung

Das derzeit genutzte Freifahrtmessgerät H39 soll um die Möglichkeit, auch Propeller mit Düse messen zu können, erweitert werden. Dafür wird ein zusätzlicher Haltearm senkrecht über der Düse angebracht, in dem eine Kraftmessdose verbaut ist. Der Haltearm ist nach Erfahrungswerten strömungsgünstig geformt, stellt aber naturgemäß eine Störung in der

Strömung dar. In der Bachelorarbeit soll untersucht werden, ob bzw. welchen hydrodynamischen Einfluss der Haltearm auf die Kräfte, genauer gesagt die Kraftmessungen hat. Diese Untersuchung soll mittels CFD (Computational Fluid Dynamics) erfolgen.

3.3 Freifahrtgerät H39

Innerhalb des Freifahrtgeräts befindet sich eine Welle, die den Propeller antreibt. Hier werden über eine Kraftmessdose das Drehmoment und der Schub gemessen. Abbildung 3.2 zeigt den prinzipiellen Aufbau eines Freifahrtgeräts mit angebrachter Düse. Das Freifahrtgerät wird stromaufwärts betrieben, der Propeller fährt also voraus und wird somit homogen angeströmt.

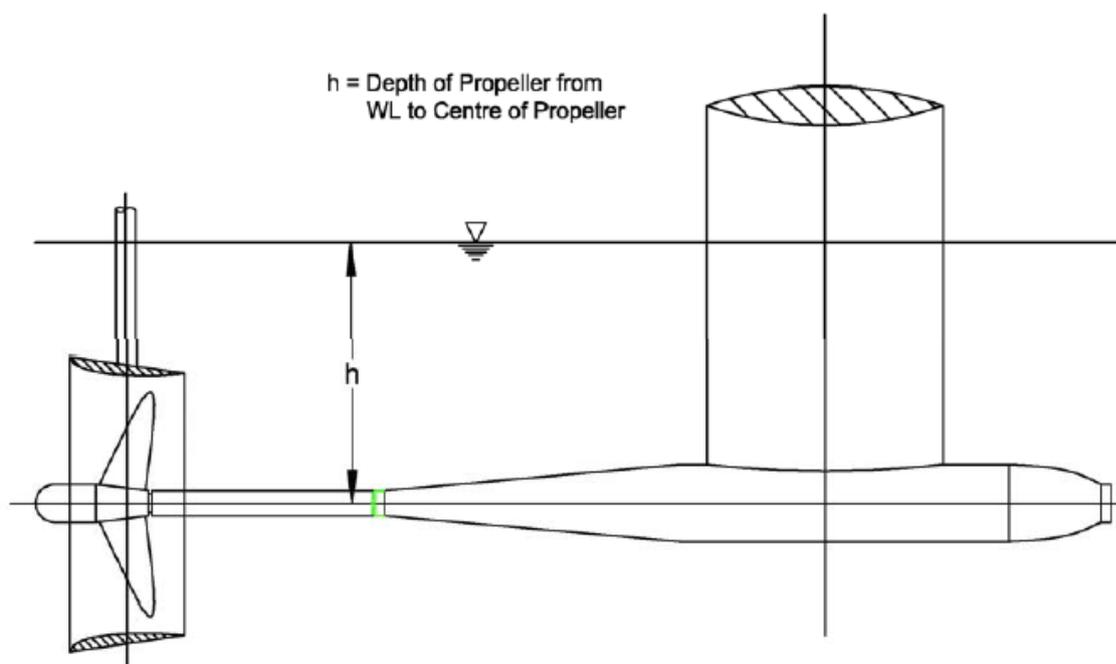


Abbildung 3.2: Schema des Freifahrtgeräts mit Düse [7]

Die Abbildung 3.3 und Abbildung 3.4 zeigen das Freifahrtgerät H32, das bereits mit einem Haltearm für die Messung mit Düsen ausgestattet ist, aber nur eingeschränkt zur Verfügung steht.

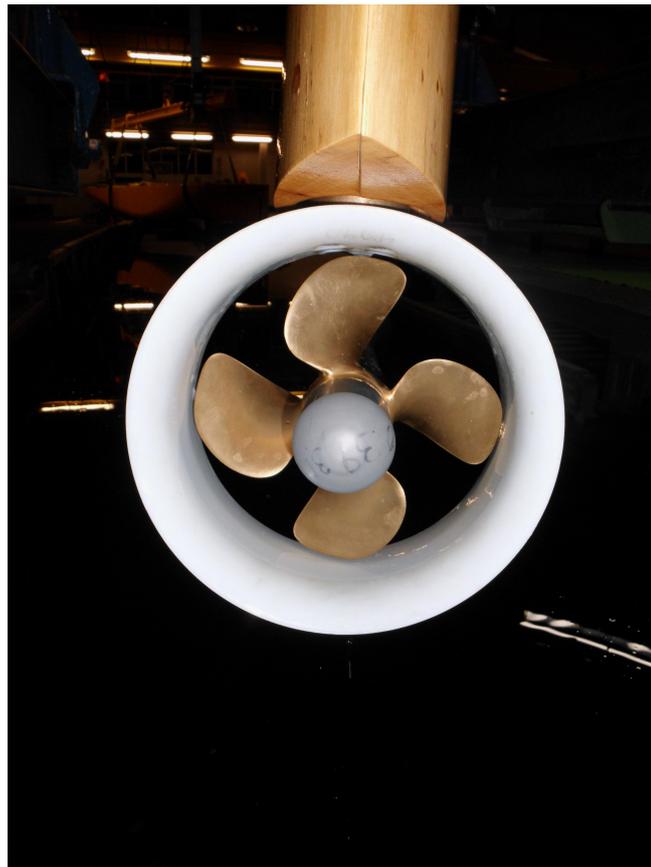


Abbildung 3.3: Freifahrtgerät H32 mit Düse frontal [7]

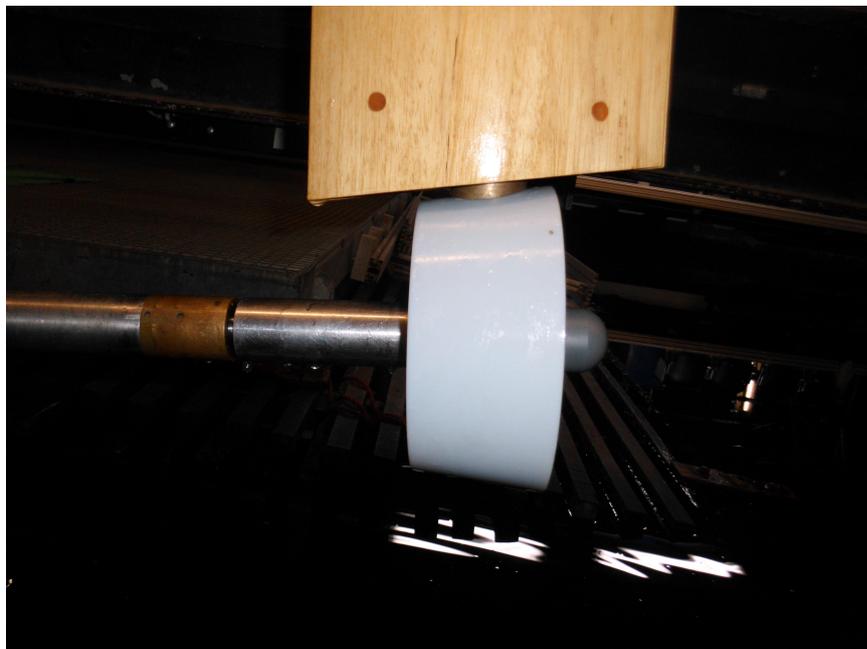


Abbildung 3.4: Freifahrtgerät H32 mit Düse [7]

4 Software

4.1 OpenFOAM

Zur Untersuchung der Problemstellung wird das CFD-Software-Paket OpenFOAM verwendet. OpenFOAM[®] (**O**pen **S**ource **F**ield **O**peration **a**nd **M**anipulation) ist ein in C++ geschriebenes Simulationssoftwarepaket, dessen Quelltext unter der „GNU General Public License“ veröffentlicht wird. Das bedeutet, der Code ist frei zugänglich und dem Nutzer steht es frei, eigene Anpassungen vorzunehmen und diese unter den gleichen Umständen zu veröffentlichen. Es ist ein Paket zum Lösen kontinuumsmechanischer Probleme, im Besonderen von Strömungsproblemen.

OpenFOAM ist eine Bibliothek, die einzelne Fähigkeiten der Software bereitstellt, wie z.B. Turbulenzmodelle und die Lösung von Differentialgleichungen. Die Eigenschaften werden für Applikationen verwendet, die in zwei Kategorien einzuordnen sind:

- **Solvers** - werden verwendet, um mathematische Problemstellungen zu lösen, wie z.B. simpleFoam.
- **Utilities** - werden für andere Aufgaben, wie Netzerstellung und Manipulation oder Vorbereitung der Simulationsdateien, verwendet.

Für die visuelle Darstellung der Ergebnisse von OpenFOAM-Simulationen wird meist die ebenfalls freie Software paraView verwendet. Diese wird bereits mit der OpenFOAM-Installation als angepasste Version namens paraFoam mitgeliefert. Alternativ kann man die Daten auch über die Kommandozeile visualisieren oder sie für andere Software exportieren.

Im Gegensatz zu den meisten kommerziellen Programmen bietet OpenFOAM keine grafische Benutzeroberfläche (GUI) und wird über die Kommandozeile und Text-Dateien gesteuert. OpenFOAM wird meistens auf Linux Distributionen verwendet, mittlerweile gibt es jedoch die Möglichkeit, OpenFOAM auch auf Windows zu installieren.

Die Vorgehensweise bei Strömungssimulationen sieht üblicherweise drei Schritte vor.

Pre-Processing

Hier werden die Einstellungen für die zu simulierende Problemstellung vorgenommen. Diese beinhalten die physikalischen Randbedingungen und Berechnungsverfahren. Außerdem gehört auch die Erstellung eines Netzes zum Pre-Processing.

Berechnung

Der vermeintlich aufwändigste Teilschritt ist die Berechnung. Für den Anwender ist bei diesem Schritt allerdings keine Handlung erforderlich.

Post-Processing

Die Aufbereitung der Daten nimmt einen weiteren Schritt in Anspruch. Diese werden je nach Anwendung tabellarisch oder visuell aufbereitet. Die Lösung wird in diesem Schritt auf Plausibilität geprüft und ggf. müssen Anpassungen vorgenommen werden.

Für die Arbeit mit OpenFOAM ist die sogenannte Case-Struktur sehr wichtig, damit die Solver und Utilities ausgeführt werden können. Das bedeutet, die Dateien und Ordner müssen wie in Abbildung 4.1 gezeigt angeordnet sein.

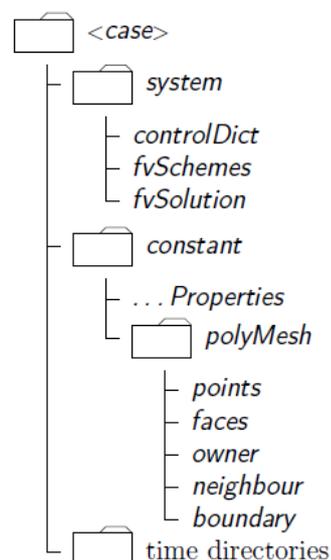


Abbildung 4.1: Case-Struktur in OpenFOAM [5]

Der system Ordner

Er enthält mindestens 3 Dateien. Die *controlDict*, die Informationen wie z.B. die Start- und Endzeit oder den Zeitschritt enthält, die *fvSchemes*, die die Diskretisierungen beschreibt und die *fvSolution*, in der die Solver und Toleranzen festgelegt werden.

Der constant Ordner

Er enthält einen weiteren Ordner namens *polyMesh*, der das Netz beschreibt und Dateien wie die *transportProperties* und die *turbulenceProperties*, wo die kinematische Viskosität ν festgelegt wird.

Die time Ordner

Er enthält vor der Simulation den Ordner 0, sofern die Simulation bei $t = 0$ startet. Hier werden die Randbedingungen und Startwerte, wie z.B. die Geschwindigkeit U oder der Druck p festgelegt. Im Laufe der Simulation werden neue Ordner mit dem entsprechenden Zeitschritt angelegt, in dem die Ergebnisse der Simulation gespeichert sind.

4.2 Vernetzer snappyHexMesh

Das Utility *snappyHexMesh* generiert dreidimensionale Gitter aus Hexaedern und erstellt damit weitgehend strukturierte Netze. Der Vorgang der Vernetzung sieht ein Hintergrundnetz aus Hexaedern vor, typischerweise wird dieses mit dem Utility *blockMesh* erstellt. An dieses Netz wird die Geometrie, vorliegend im Stereolithographie (STL) oder Wavefront Object (OBJ) Format, mittels Zellteilung angepasst. Die überflüssigen Zellen werden danach gelöscht. Anschließend müssen die Zellen noch an die exakte Kontur der Geometrie angepasst werden, dies geschieht im Schritt *snapping*. Als letztes werden die Layer hinzugefügt. Hier wird das Netz etwas von der Oberfläche weggeschoben und die Layer werden direkt auf der Oberfläche aufgebracht.

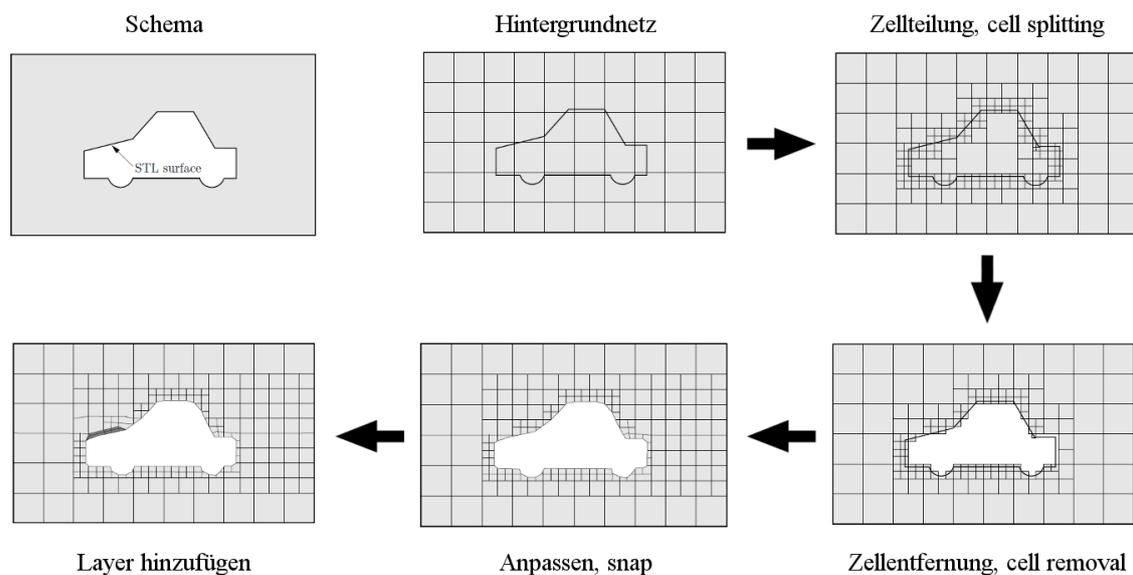


Abbildung 4.2: Schema der Vernetzung mit *snappyHexMesh* (in Anlehnung an[5])

4.3 Post-Processing mit ParaView/paraFoam

Die mit OpenFOAM mitgelieferte Post-Processing-Software ist ParaView. Diese ist ebenso wie OpenFOAM Open Source und frei verfügbar. Aufgerufen wird sie für OpenFOAM mit

dem Befehl

`paraFoam`

welcher ein Skript ausführt, um eine eigens für OpenFOAM angepasste Version von ParaView auszuführen.

5 Diskretisierung und Numerik

5.1 Theoretische Grundlagen zur Strömungsberechnung

Die theoretischen Grundlagen sind in Standard-Literaturen zu den Themen Strömungsmechanik und numerischer Strömungsmechanik nachzulesen. In diesem Abschnitt sollen die grundlegenden Erhaltungsgleichungen und Ansätze vorgestellt werden. Den Beschreibungen liegen vor allem die Werke von Ferziger und Perić [4], Laurien und Oertel jr. [13], M. Böhle, H. Oertel jr., T. Reviol [1] und die Vorlesungsunterlagen von Prof. Dr.-Ing. Wulf von der Hochschule für angewandte Wissenschaften Hamburg [16] zugrunde.

Massenerhaltung

In ingenieurmäßigen Anwendungen gilt, dass Masse weder erzeugt noch vernichtet werden kann. Dies lässt sich mit

$$\frac{dm}{dt} = 0 \quad (5.1)$$

beschreiben. Hier beschreiben m eine Kontrollmasse (KM) und t die Zeit. Dies wird in eine Integralform für Kontrollvolumen (KV) umgeformt und man erhält

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \mathbf{v} \cdot \mathbf{n} dS = 0 \quad (5.2)$$

wobei V das Volumen, ρ die Dichte, S die Oberfläche, \mathbf{v} den Geschwindigkeitsvektor mit den Komponenten u_x, u_y, u_z und \mathbf{n} den Einheitsvektor senkrecht zur Oberfläche des KV beschreiben.

Wendet man den Gaußschen Divergenzansatz auf die Gleichung an, erhält man eine koordinatenfreie Differentialform der Kontinuitätsgleichung

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (5.3)$$

Für inkompressible Fluide vereinfacht sich der Term in kartesischen Koordinaten zu

$$\frac{\partial u_j}{\partial x_j} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \quad (5.4)$$

Navier-Stokes-Gleichungen

Das zweite Newtonsche Axiom besagt, die zeitliche Änderung eines Impulses ist gleich der auf einen Körper wirkenden Kräfte. Dies lässt sich als

$$\frac{d(m \cdot \mathbf{v})}{dt} = \sum \mathbf{F} \quad (5.5)$$

schreiben, wobei \mathbf{F} die auf die Kontrollmasse wirkende Kraft ist. Formt man auch diese Gleichung für die Kontrollvolumenmethode um, erhält man

$$\frac{\partial}{\partial t} \int_V \rho \mathbf{v} dV + \int_S \rho \mathbf{v} \mathbf{v} \cdot \mathbf{n} dS = \sum \mathbf{F} \quad (5.6)$$

Um die extensive Größe \mathbf{F} mit intensiven Eigenschaften auszudrücken, werden die Kräfte, die auf ein KV wirken, betrachtet. Dies sind

- Oberflächenkräfte, z.B. Druck, Scherspannungen oder Oberflächenspannungen
- Volumenkräfte, z.B. Gravitation, Zentrifugal- und Corioliskräfte

Für newtonsche Fluide kann der Spannungstensor T in Indexschreibweise und karthesischen Koordinaten als

$$T_{ij} = - \left(p + \frac{2}{3} \mu \frac{\partial u_j}{\partial x_j} \right) \delta_{ij} + 2\mu D_{ij} \quad (5.7)$$

mit

$$D_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \quad (5.8)$$

geschrieben werden. Dabei steht μ für die dynamische Viskosität, p für den statischen Druck, D für den Tensor der Deformationsrate und δ_{ij} für das Kroenecker Symbol ($\delta_{ij} = 1$ wenn $i = j$ sonst $\delta_{ij} = 0$).

Dies führt zur Integralform der generischen Transportgleichung

$$\underbrace{\frac{\partial}{\partial t} \int_V \rho \phi dV}_{\text{instationärerTerm}} + \underbrace{\int_S \rho \phi \mathbf{v} \cdot \mathbf{n} dS}_{\text{konvektiverTerm}} = \underbrace{\int_S \Gamma \nabla \phi \cdot \mathbf{n} dS}_{\text{Diffusionsterm}} + \underbrace{\int_V q_\phi dV}_{\text{Quellterm}} \quad (5.9)$$

wobei Γ den Diffusionskoeffizienten der generischen Größe ϕ darstellt.

Beschränkt man sich auf Newtonsche Fluide $\mu \neq f(\tau)$ und inkompressible Strömungen, ergeben sich die Navier-Stokes-Gleichungen (NS-Gleichungen) zu

$$\rho \cdot \left(\frac{\partial u_x}{\partial t} + u_x \cdot \frac{\partial u_x}{\partial x} + u_y \cdot \frac{\partial u_x}{\partial y} + u_z \cdot \frac{\partial u_x}{\partial z} \right) = k_x - \frac{\partial p}{\partial x} + \mu \cdot \left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right) \quad (5.10)$$

$$\rho \cdot \left(\frac{\partial u_y}{\partial t} + u_x \cdot \frac{\partial u_y}{\partial x} + u_y \cdot \frac{\partial u_y}{\partial y} + u_z \cdot \frac{\partial u_y}{\partial z} \right) = k_y - \frac{\partial p}{\partial y} + \mu \cdot \left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \right) \quad (5.11)$$

$$\rho \cdot \left(\frac{\partial u_z}{\partial t} + u_x \cdot \frac{\partial u_z}{\partial x} + u_y \cdot \frac{\partial u_z}{\partial y} + u_z \cdot \frac{\partial u_z}{\partial z} \right) = k_z - \frac{\partial p}{\partial z} + \mu \cdot \left(\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \right) \quad (5.12)$$

wobei $\mathbf{k} = (k_x, k_y, k_z)$ den Volumenkräften (und damit meist der Gewichtskraft, also $\mathbf{k} = \rho \cdot \mathbf{g}$) mit $\mathbf{g} = (g_x, g_y, g_z)$) entspricht.

In koordinatenfreier Vektorschreibweise kann man dies auch mit

$$\rho \cdot \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = \mathbf{k} - \nabla p + \mu \cdot \Delta \mathbf{v} \quad (5.13)$$

darstellen. Mit den Navier-Stokes-Gleichungen und der Kontinuitätsgleichung stehen nun vier partielle Differentialgleichungen zur Verfügung, um die Unbekannten u_x, u_y, u_z und p zu bestimmen. Diese Gleichungen werden je nach Bedarf auch noch um die Energiegleichung erweitert.

RANS-Gleichungen

Da eine analytische Lösung der NS-Gleichungen nur in Ausnahmefällen möglich ist, müssen diese Gleichungen vereinfacht werden. Betrachtet man inkompressible Strömungen und wendet den Mittelungssatz von Reynolds mit folgenden Ansätzen an, ergibt sich:

$$\begin{aligned} u_x &= \overline{u_x} + u'_x, & u_y &= \overline{u_y} + u'_y, \\ u_z &= \overline{u_z} + u'_z, & p &= \overline{p} + p' \end{aligned} \quad (5.14)$$

wobei der Überstrich \overline{f} den zeitlichen Mittelwert und der einfach gestrichene Term f' die Abweichung davon bezeichnet.

Setzt man diesen Ansatz in die Erhaltungsgleichungen ein, erhält man die Reynoldsgemittelten-Navier-Stokes-Gleichungen (**R**eynolds- **A**veraged-**N**avier-**S**tokes-Equations)

$$\frac{\partial(\rho \overline{\mathbf{v}})}{\partial t} + \nabla(\rho \overline{\mathbf{v}} \cdot \overline{\mathbf{v}}) + \underbrace{\overline{\rho \mathbf{v}' \mathbf{v}'}}_{\text{Reynolds-
spannungen}} = \mathbf{k} - \nabla p + \mu \cdot \Delta \mathbf{v} \quad (5.15)$$

und die Reynoldsspannungen werden zu:

$$\overline{\rho \mathbf{v}' \mathbf{v}'} = \rho \begin{pmatrix} \overline{u'^2} & \overline{u'_x u'_y} & \overline{u'_x u'_z} \\ \overline{u'_x u'_y} & \overline{u'^2_y} & \overline{u'_y u'_z} \\ \overline{u'_z u'_x} & \overline{u'_z u'_y} & \overline{u'^2_z} \end{pmatrix} \quad (5.16)$$

Also kommen zu den 4 unbekanntten Größen noch 6 unbekannte Spannungen hinzu, jedoch stehen nur 4 Gleichungen zur Verfügung. Dies wird als Schließungsproblem bezeichnet. Um die Gleichungen zu schließen, werden Turbulenzmodelle eingeführt.

Turbulenzmodelle

Die meisten in der Technik verwendeten RANS-Modelle basieren auf dem Wirbel-Viskositäts Ansatz nach Boussinesq. Demnach sind die Reynolds-Spannungen proportional zum Schergradienten und es werden zusätzliche Größen wie die Turbulenzviskosität μ_t und die turbulente, kinetische Energie k eingeführt. Zu den RANS-Modellen gehören folgende Ansätze:

- Einpunktschließungsansätze
 - Null-Gleichungsmodelle
 - Ein-Gleichungsmodelle
 - Zwei-Gleichungsmodelle
- Zweipunktschließungsansätze
- Reynoldsspannungsmodelle

In der Ingenieurspraxis werden vorwiegend die Einpunktschließungsansätze verwendet. Oft verwendet werden beispielsweise das Spalart-Allmaras-Modell (Ein-Gleichungsmodell), oder die Zwei-Gleichungsmodelle wie das k - ϵ -Modell und das k - ω -Modell und deren Derivate. Außerdem können die Simulationen auch über die sogenannten *Large Eddy Simulations (LES)* oder die *Direkte Numerische Simulation (DNS)* gelöst werden. Der Grad der Vereinfachungen sinkt in dieser Reihenfolge, jedoch steigt der numerische Aufwand dabei immens, was diese Ansätze für technische Belange noch unbrauchbar macht. Mit steigender Rechenkapazität kann sich dies in den nächsten Jahren ändern.

Finite-Volumen-Methode

In der CFD werden die Grundgleichungen der Strömungsmechanik durch Approximation gelöst. Die Differentialgleichungsterme können z.B. über Ansätze wie die Finite-Elemente-Methode (FEM), die Finite-Differenzen-Methode (FDM) oder eben die Finite-Volumen-Methode (FVM) gelöst werden. Die Finite-Volumen-Methode beschreibt ein Diskretisierungskonzept. Diskretisierung bedeutet hier, dass die kontinuierlich formulierten Differentialgleichungen (DGL) auf diskrete Variablen an ausgewählten Raumpunkten eines Rechengitters anzuwenden. Die Raumpunkte sind durch das Rechengitter definiert und die Variablenwerte an den Raumpunkten sind die zu berechnenden Unbekannten. Diese Unbekannten sollen in ein algebraisches Gleichungssystem überführt und damit iterativ bestimmt werden.

Bei der Finite-Volumen-Methode wird das Berechnungsgebiet in einzelne KV, also Zellen aufgeteilt. Innerhalb dieser Zellen wird eine Größe als Mittelwert aufgefasst und für jedes KV wird die Gleichung 5.9 angewendet. Die Oberflächenintegrale ermöglichen die Bilanzierung zwischen benachbarter Zellen wie z.B. den Massenfluss. Unter Beachtung der Randbedingungen ergibt sich ein Gleichungssystem der Form

$$a_P \phi_P = \sum_{NB} a_{NB} \phi_{NB} + b \quad (5.17)$$

mit den Indizes $(\dots)_P$ für den Zell- oder Flächenmittelpunkt und $(\dots)_{NB}$ für die Nachbarzelle. a und b stehen hier für die Variablen des Systems, die sich aus dem Diskretisierungsansatz ergeben.

5.2 Netz-Generierung

Wie im Kapitel Software bereits erwähnt, wird das Netz mithilfe eines Hintergrundnetzes generiert. Dieses wird durch ein *blockMeshDict* beschrieben und enthält erste Informationen über die Feinheit des Netzes. Es wurde ein Grading eingesetzt, das die Bereiche an den Ein- und Ausströmändern gröber vernetzt und den Bereich um den Haltearm des Freifahrtgeräts feiner auflöst. Anschließend wird mithilfe des *snappyHexMeshDicts* das finale Netz erzeugt. Hier werden Einstellungen zum genauen Aufbau festgelegt. In dieser Datei werden die STL-Dateien eingelesen und in das *blockMesh* eingefügt. An dieser Stelle ist es ebenso möglich, Bereiche für Verfeinerungen (wie eine Box oder eine Kugel) festzulegen. In diesem Fall wurden auch *surfaceRefinements*, also Oberflächenverfeinerungen, verwendet. Eine Verfeinerung um 1 bedeutet, dass die Zellen in diesem Bereich durch vier geteilt werden. Im letzten Schritt werden die Layer hinzugefügt, dies geschieht unter anderem mittels der Funktion *firstLayerThickness* und dem *ExpansionRatio*. Zu beachten ist hierbei, dass es sich um relative Werte handelt.

5.3 Randbedingungen

Die Randbedingungen für die Berechnung des Haltearms werden für den Einlass namens *inlet*, den Auslass *outlet*, die domainbegrenzenden Wände *wand*, die Symmetrie-Ebene *symmetrie* und den Haltearm selbst, bestehend aus *duese*, *schwert* und *halter* gesetzt.

fixedValue

Die *fixedValue* Bedingung beschreibt einen konstanten Wert, der jeder Zelle des Randes zugewiesen wird. Am Einlass wird der Geschwindigkeit und am Auslass dem Druck ein fester Wert vorgegeben. Auch die turbulente, kinetische Energie k und die turbulente Frequenz ω werden am *inlet* auf einen festen Wert gesetzt, wobei diese als numerische Randbedingungen zu sehen sind. Dieser feste Wert entspricht einer Dirichlet-Randbedingung.

zeroGradient

Diese trägt einen Null-Gradienten des *internalFields* auf den Rand auf. Diese Bedingung kommt für den Druck sowohl am *inlet* als auch für die reibungsbehafteten Wände zum Einsatz.

noSlip

Diese Randbedingung ist äquivalent zu einer *fixedValue* Bedingung die auf den Wert *fixedValue uniform (0 0 0)* gesetzt wurde. Sie kommt für Wände, an denen Haftung

herrscht, zum Einsatz.

inletOutlet

An sich ist die `inletOutlet` Einstellung gleich dem `zeroGradient`, sie wechselt jedoch zum `fixedValue`, wenn der Geschwindigkeitsvektor in die Domain zeigt, also im Falle einer Rückströmung.

calculated

Hier wird die Randbedingung von OpenFOAM berechnet. Diese Einstellung kommt nur bei der turbulenten Scheinviskosität ν_t zum Einsatz.

symmetry

Die `symmetry` ist eine Randbedingung, die geometrischer Natur ist und eine Symmetrie beschreibt. Numerisch kommt hier die `slip` Bedingung zum Einsatz. Diese stellt sich als `zeroGradient` ein, wenn ϕ ein Skalar ist. Bei einem Vektor wird die normalen Komponente zu `fixedValue` von null und die tangentialen Komponenten werden zum `zeroGradient`.

5.4 Turbulenzmodell

Für die Berechnungen wurde das *k- ω -SST-Modell* verwendet. Dieses Modell vereinigt die Stärken des *k- ϵ -Modells* und des *k- ω -Modells* und wurde 1993 von Florian R. Menter vorgestellt. Es handelt sich um ein Zwei-Gleichungsmodell. Das Shear-Stress-Transport(SST)-Modell basiert auf der Annahme, dass die Schubspannung in den logarithmischen und den Nachstrombereichen der turbulenten Grenzschichten proportional zu der turbulenten, kinetischen Energie k ist. In den Bereichen der Grenzschichten entspricht das SST-Modell dem *k- ω -Modell*, erweitert um die eben genannte Annahme und in der freien Strömung findet das *k- ϵ -Modell* Anwendung.[14]

5.5 Wandfunktionen

kqRWallFunction

Die *kqRWallFunction* ist eine *zeroGradient* Bedingung für die Werte von k , Q und R .

omegaWallFunction

Diese Wandfunktion beschreibt ω mit einem glatten Übergang nach

$$\omega = \sqrt{\omega_{vis}^2 + \omega_{log}^2} \quad (5.18)$$

mit ω_{vis} in der viskosen Schicht und ω_{log} in der logarithmischen Region.

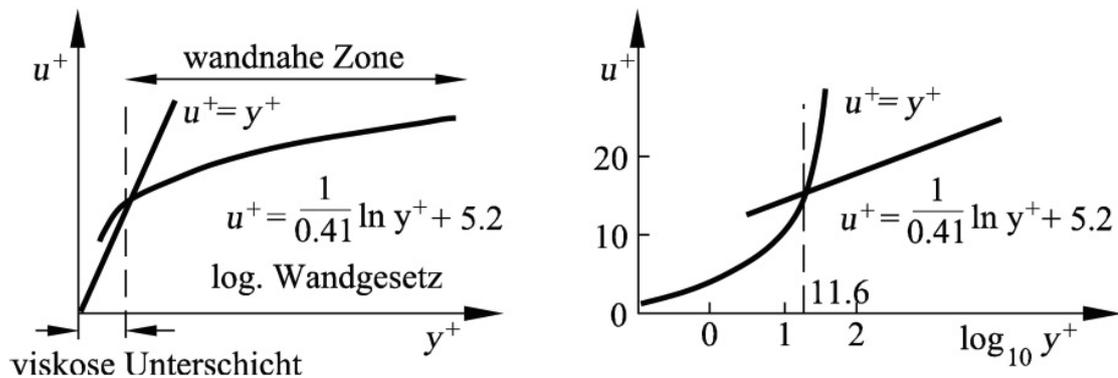


Abbildung 5.1: Geschwindigkeitsprofile in der viskosen Unterschicht und der wandnahen Schicht [13]

nutkWallFunction

In dieser Wandfunktion wird die Berechnung von dem dimensionslosen Wandabstand y^+ auf

$$y^+ = \frac{C_\mu^{0.25} \cdot y \cdot \sqrt{k}}{\nu} \quad (5.19)$$

festgelegt. Hierbei sind C_μ eine Konstante und y der Wandabstand.

5.6 Approximationen

steadyState

Dieses Schema gehört zu den Zeitableitungen und gibt 0 aus.

$$\frac{\partial}{\partial t}(\phi) = 0 \quad (5.20)$$

Es bestimmt also einen stationären Fall.

bounded Gauss upwind

bounded Gauss upwind ist ein Aufwindverfahren erster Ordnung. Die Lösungen sind im allgemeinen diffusiv und damit ungenau. Allerdings gelten sie auch als sehr stabil.

bounded Gauss linearUpwind

Dieses Schema ist ein Aufwindverfahren zweiter Ordnung, also eine lineare Interpolation. Dies ist die einfachste Form mit einer Genauigkeit zweiter Ordnung und bietet somit einen guten Kompromiss zwischen Genauigkeit und Aufwand (vgl. [4] S. 104).

Gauss linear

Der Eintrag `Gauss linear` legt die Finite-Volumen-Diskretisierung nach Gauss fest. Der Zusatz `linear` steht für die lineare Interpolation.

5.7 Lösungsalgorithmen

Zum Lösen der Gleichungssysteme stehen verschiedene Lösungsansätze zur Verfügung. Für die NS-Gleichungen muss nicht nur die Geschwindigkeit, sondern auch der Druck bestimmt werden. Dies wird meist über iterative Verfahren indirekt über die Massenerhaltungsgesetze realisiert. Zu nennen sind hier Solver wie der SIMPLE oder PISO Solver. Da in dieser Arbeit der SIMPLE-Algorithmus zum Einsatz kam, soll dieser kurz beschrieben werden.

SIMPLE Algorithmus

Der SIMPLE Algorithmus (**S**emi **I**mplicit **M**ethod for **P**ressure **L**inked **E**quations) dient der Druck-Geschwindigkeits-Kopplung. Der Druck muss die Kontinuitätsgleichung erfüllen – zeitgleich sind die Druckberechnung und die Kontinuitätsgleichung miteinander gekoppelt. Dies führt zur semi-impliziten Behandlung der Terme, wobei ein Teil zum vorigen und ein Teil zum jetzigen Zeitpunkt behandelt wird. Dazu wird wie folgt vorgegangen: Der Druck wird zunächst geschätzt. Daraus werden die Geschwindigkeiten aus der Impulsgleichung berechnet. Anhand der Kontinuitätsgleichung wird der Druck korrigiert und damit auch die Geschwindigkeiten korrigiert. Nun wird die Impulsgleichung nicht mehr erfüllt und deswegen wird neu iteriert, bis beide Gleichungen (annähernd) erfüllt sind.

$$\begin{aligned} p^m &= p^* + p', & u_x^m &= u_x^* + u'_x, \\ u_y^m &= u_y^* + u'_y, & u_z^m &= u_z^* + u'_z \end{aligned} \quad (5.21)$$

Hier sind die Indizes $(\dots)^m$ der Zähler für die äußere Iteration, der $(\dots)^*$ der Schätzwert und $(\dots)'$ der Korrekturwert.

Anschaulich lässt sich der Verlauf des SIMPLE Verfahrens mit einem Flussdiagramm wie folgt darstellen:

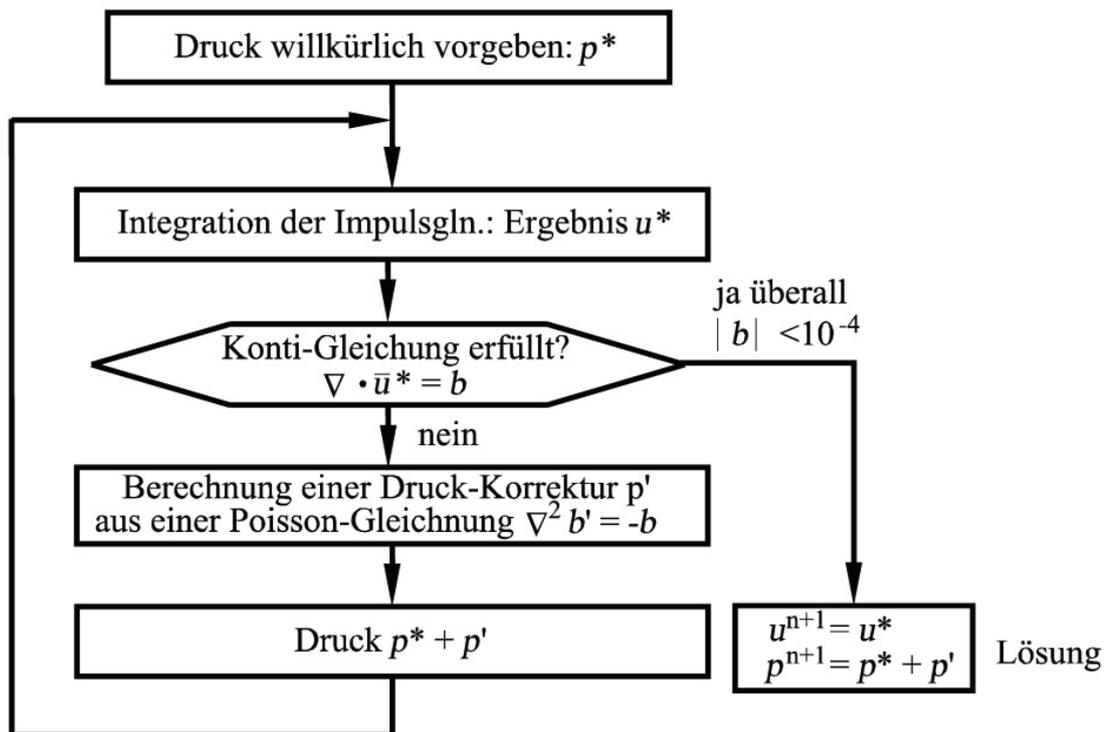


Abbildung 5.2: Schema zur Berechnung des Druckes und Erfüllung der Kontinuitätsgleichung nach dem SIMPLE-Verfahren [13]

Es gibt auch Abwandlungen des SIMPLE-Verfahrens wie das SIMPLER- oder SIMPLEC-Verfahren. Beispielsweise das SIMPLEC- bzw. SIMPLE Consistent Verfahren konvergiert bei einfachen Problemen schneller.

6 Simulation des Freifahrtgeräts

Um den Einfluss des Haltearms einschätzen zu können, wird zu diesem Zweck die gleiche Simulation mit und ohne Haltearm durchgeführt. Im Folgenden sollen die Einstellungen der Fälle vorgestellt werden. Sämtliche für die Berechnung relevante Dateien sind in Anhang A zu finden.

6.1 Gitteraufbau

Das Erstellen des Gitters wurde bereits in Abschnitt 4.2 im Groben beschrieben.

blockMesh

Hierzu wird zuerst ein block-strukturiertes Netz mit den Abmessungen des gewünschten Berechnungsgebietes erstellt. Dies geschieht mithilfe des *blockMeshDicts* und der Funktion `blockMesh`. Die Feinheit des blockMeshes ist für die weitere Vernetzung maßgeblich. Aus diesem Grund wurde angestrebt, die Abmessungen der Volumina würfelförmig mit einer Kantenlänge von 20 mm zu gestalten. Da die Berechnungszeiten von der Zellanzahl abhängig sind, wurde diese Kantenlänge nur in unmittelbarer Nähe zur Position der Geometrie verwirklicht und die Kantenlängen werden mithilfe eines `simpleGrading` zu den Rändern hin größer. Um dies zu realisieren, mussten 3 Blöcke angelegt werden. Tabelle 6.1 zeigt, welche Werte den Blöcken zugeordnet werden.

	Zellen	x	y	z	Grading	x	y	z
Block Mitte	35	25	70		0,2	2,14	1	
Block Inlet	48	25	70		1	2,14	1	
Block Outlet	64	25	70		5,02	2,14	1	

Tabelle 6.1: `simpleGrading blockMeshDict`

Tabelle 6.2 zeigt die Abmessungen der gesamten Domain, die im *blockMeshDict* angegeben wird.

Wie die Blöcke angeordnet sind, veranschaulicht Abbildung 6.1. Zu beachten ist hier, dass die Zahlen des mittleren Blocks schlecht lesbar sind, da die Eckpunkte im *blockMeshDict* doppelt belegt wurden. Daher sind hier zwei Zahlen auf einer Stelle zu sehen.

	x_{min} [mm]	y_{min} [mm]	z_{min} [mm]	x_{max} [mm]	y_{max} [mm]	z_{max} [mm]
Block Inlet	200	-750	-680	2600	0	709
Block Mitte	-500	-750	-680	200	0	709
Block Outlet	-3700	-750	-680	-500	0	709

Tabelle 6.2: Abmessungen blockMesh

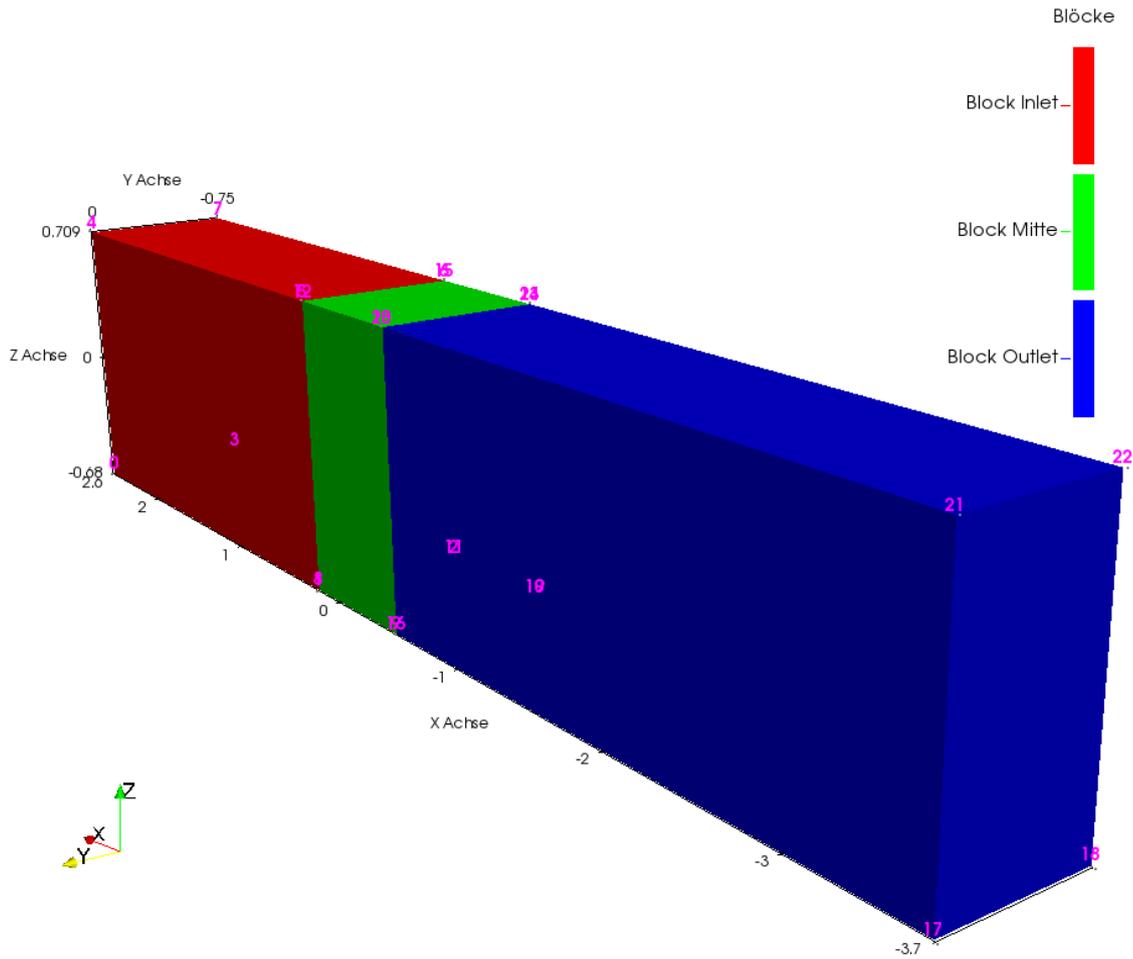


Abbildung 6.1: blockMesh Ansicht

Aus dem User Guide [5] kann man entnehmen, wie die Expansion ratios bestimmt werden (hier als Grading oder Verfeinerung bezeichnet). Der Wert beschreibt das Verhältnis von der letzten zur ersten Zelle.

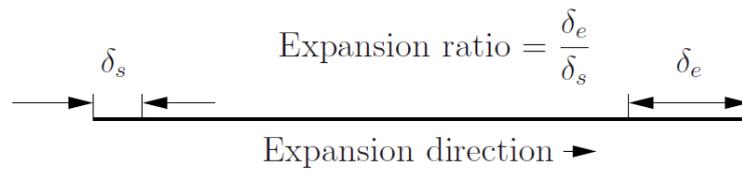


Abbildung 6.2: Netzverfeinerung an einer Blockkante [5]

Um eine einfache Handhabung dieser Werte zu erreichen, wurde ein Online-Tool namens *blockMesh grading calculation* verwendet.

Scripts/blockMesh grading calculation

< Scripts

The purpose of this script is to calculate the grading factor as described in the [user guide](#).

- Three of the 6 values can be specified. The other 3 values are calculated from them. Specify the input values with the three drop-down buttons.
- Not all combinations of input values are valid (for instance: there has to be at least one length specified to calculate the others)

Width of end cell	Total length	Number of cells
Total length	2400	
Number of cells	48	
Total expansion ratio	0,1995323417	
Cell-to-cell expansion ratio	0,9662881774	
Width of start cell	100,2343772	
Width of end cell	20	

Abbildung 6.3: blockMesh grading calculation [6]

In dem gezeigten Beispiel des Rechners sieht man die Werte in x-Richtung des mittleren Blocks. Der *Total expansion ratio* ergibt sich zu 1,995, was zu 0,2 aufgerundet wurde. Analog wurden die anderen Werte ermittelt.

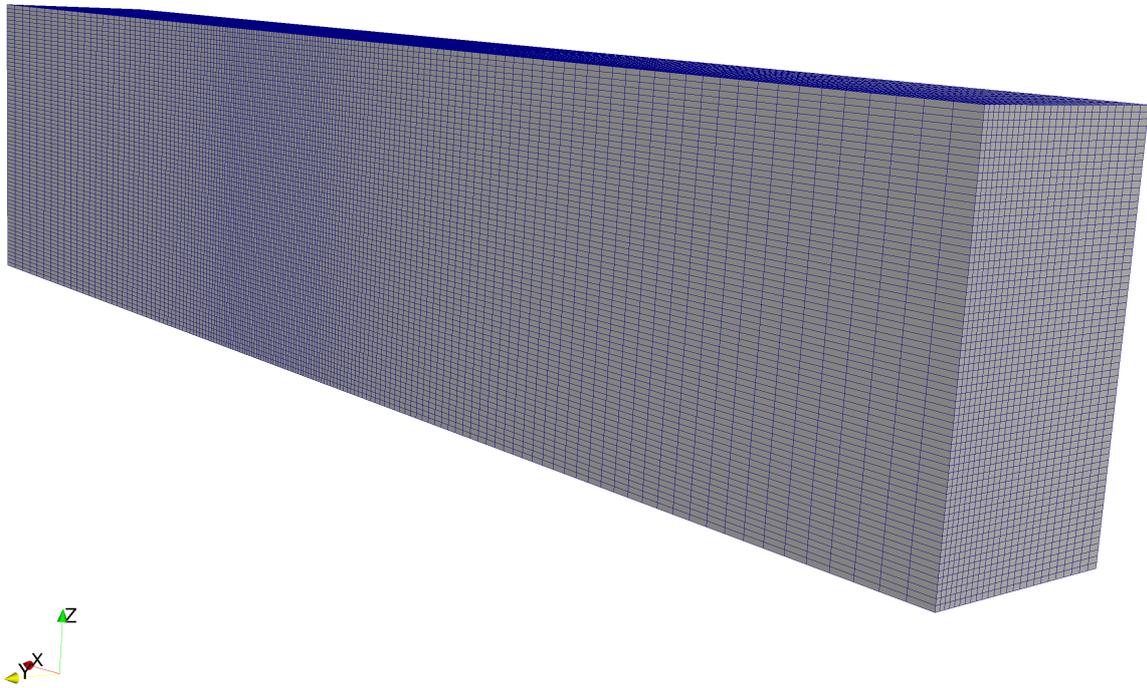


Abbildung 6.4: blockMesh Gesamtansicht

In Abbildung 6.4 erkennt man das Grading zur Mitte hin sowie die länglichen Randzellen und in Abbildung 6.5 sieht man die quadratischen Zellen, die zu den Seiten ebenso länglich werden.

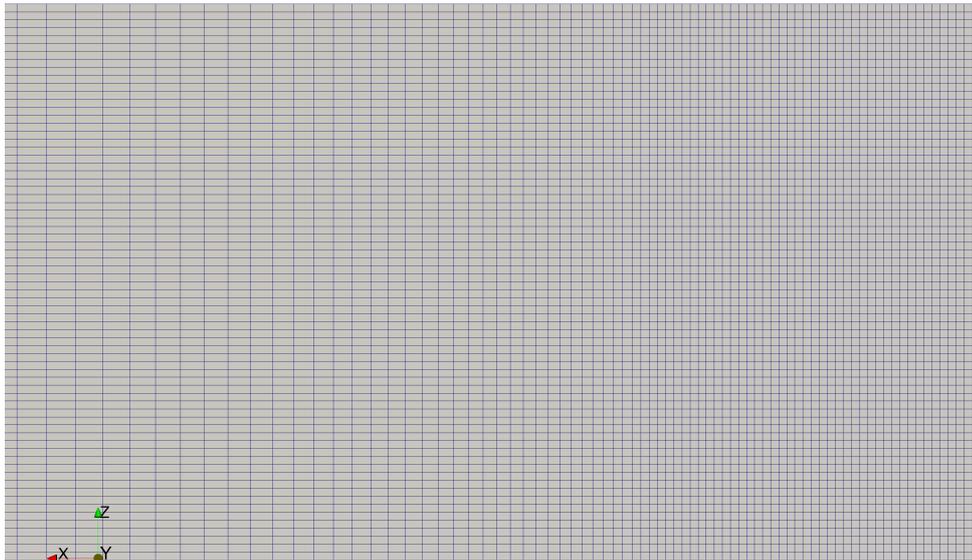


Abbildung 6.5: blockMesh Detail

snappyHexMesh

Im *snappyHexMeshDict* werden die drei Schritte *castellatedMesh*, *snap* und *addLayers* spezifiziert. In diesem Fall ist zu beachten, dass keine Layer hinzugefügt werden. Deswegen ist die Einstellung auf `false` gesetzt. Dies macht die Angaben zur Layergenerierung überflüssig, sie sind jedoch der Vollständigkeit halber in der Dictionary-Datei verblieben. In der Kategorie *geometry* werden einerseits die Geometrien im STL-Format eingelesen und andererseits Verfeinerungsgeometrien erzeugt. Die Geometrie wird in diesem Fall im Koordinatenursprung in das *blockMesh* eingefügt. Abbildung B.1 zeigt die Geometrie des Haltearms. Dies können Zylinder, Quader, Kugeln usw. sein. Hier sollen ein Zylinder den Bereich um die Düse und den weiteren Verlauf stromabwärts, eine (Halb-)Kugel den kleinen Halter zwischen Düse und Schwert und ein Quader den Bereich der Unterkante des Schwertes weiter auflösen. Eingangs werden Variablen für die Verfeinerung angegeben. Es kann zwischen `refinementSurfaces` und den `refinementRegions` unterschieden werden. Die `refinementSurfaces` verfeinert die Oberflächen der Geometrien, während die `refinementRegions` die angegebenen Geometrien regionsweise verfeinert. Wie anfangs in Abschnitt 6.1 erwähnt, ist die Größe der im *blockMeshDict* festgelegten Zellgrößen nun entscheidend. Die Verfeinerungen werden nicht absolut, sondern relativ angegeben. Denn ein Verfeinerungsgrad von 1 entspricht dem Teilen einer Zelle je einmal in jede Raumrichtung. Das bedeutet, dass aus einer Zelle, die um 1 verfeinert wird, 8 neue Zellen entstehen.

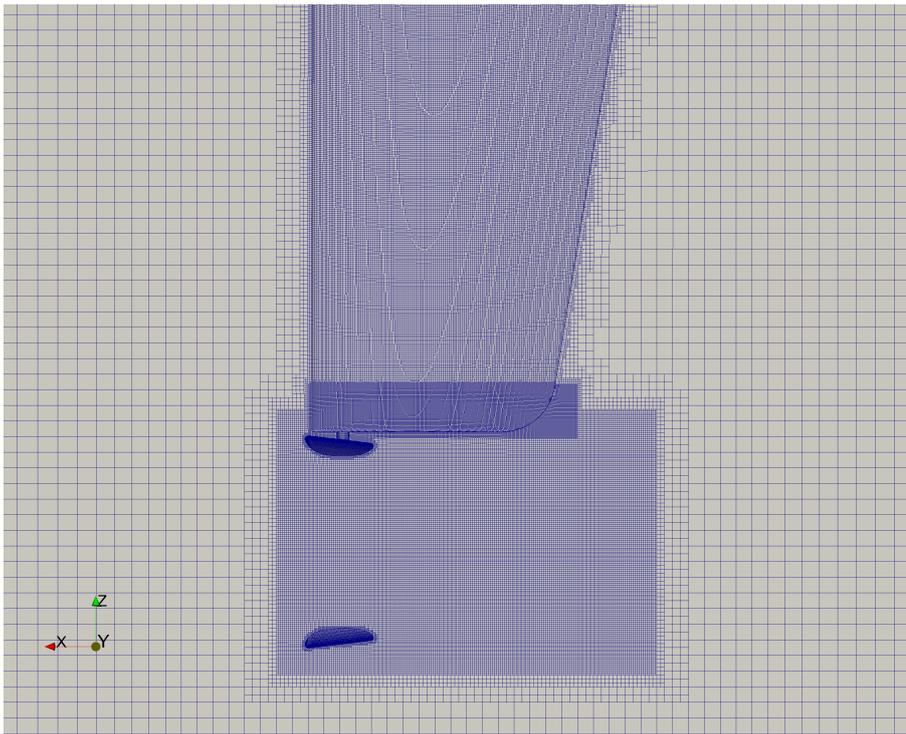


Abbildung 6.6: Ausschnitt des Netzes an der Geometrie

In der Gesamtansicht sieht man den Zylinder um die Düse in Form eines Rechteckes, da die Ansicht in der Symmetrieebene liegt. Auch den Quader um die Unterkante des Schwertes erkennt man gut. Die Verfeinerungen der Oberflächen sind auch zu sehen.

Am Bereich der Halterung erkennt man in Abbildung 6.7 die kreisförmige Verfeinerung der Kugel nicht, da die Verfeinerungen der Kugel und der Umgebung gleich sind. Auf weitere Verfeinerungen wurde verzichtet, da dies zu kleineren y^+ -Werten geführt hätte. Problematisch bei der Vernetzung waren vor allem die scharfen Kanten. Diese führen bei snappyHexMesh zu Verzerrungen im Gitter. In dem finalen Netz konnten die Verzerrungen auf die Hinterkante beschränkt und minimalisiert werden. Sie genügten schließlich den Qualitätsanforderungen und wurden als weniger relevant betrachtet, da der Einfluss weit genug von der Kraftmessung entfernt sein würde. Verfeinerungen hätten das Netz verbessert, aber auch einen Einfluss auf die Auflösung in Wandnähe gehabt, weswegen darauf verzichtet und die leichten Verzerrungen hingenommen wurden.

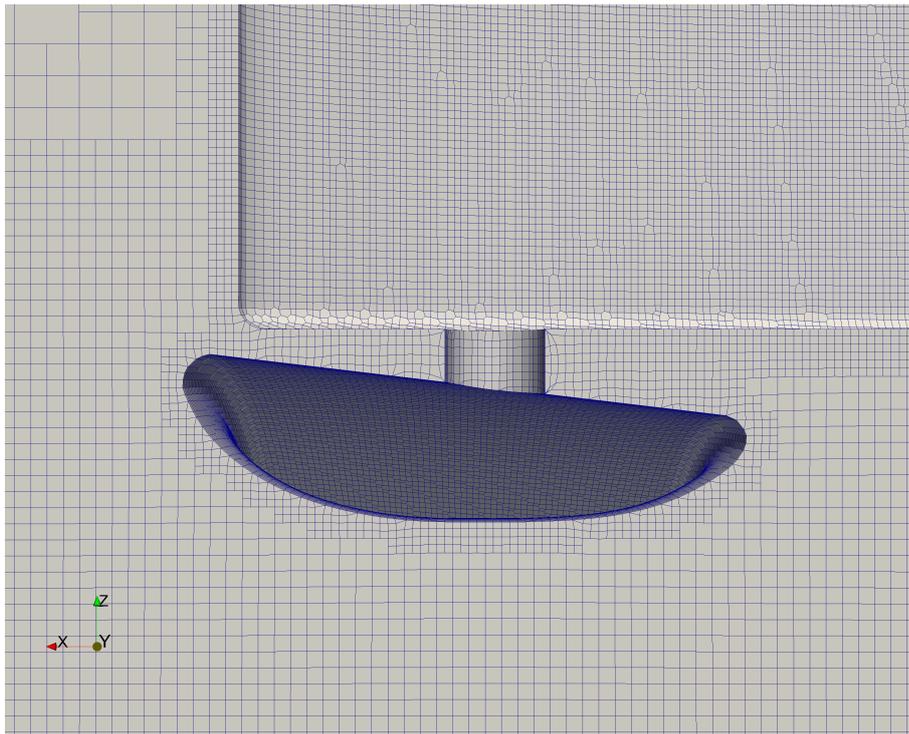


Abbildung 6.7: Ausschnitt des Netzes an der Halterung

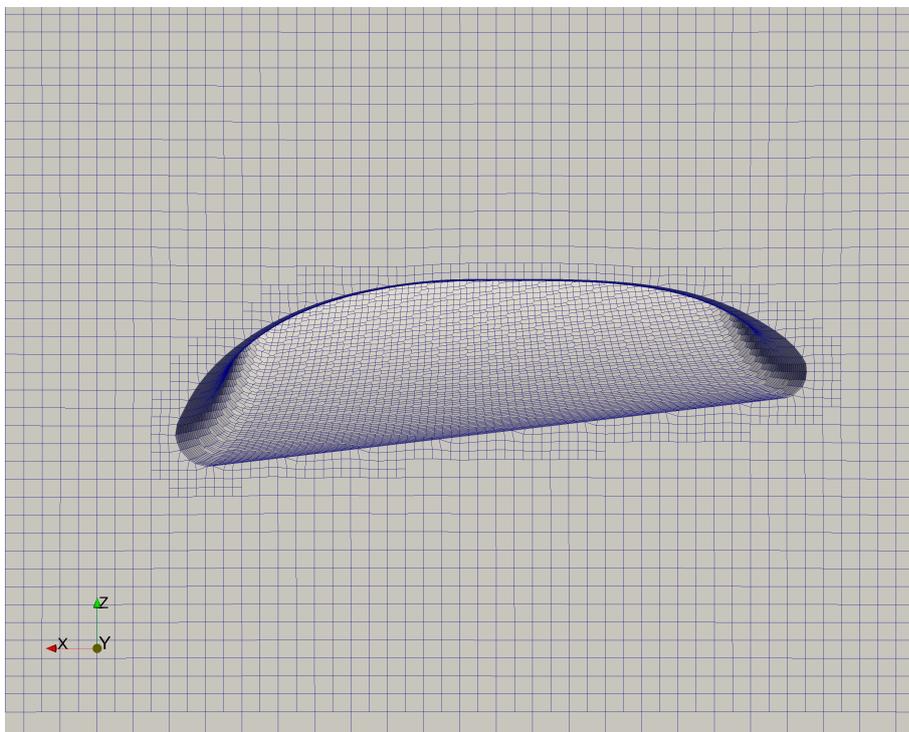


Abbildung 6.8: Ausschnitt des Netzes an Düse unten

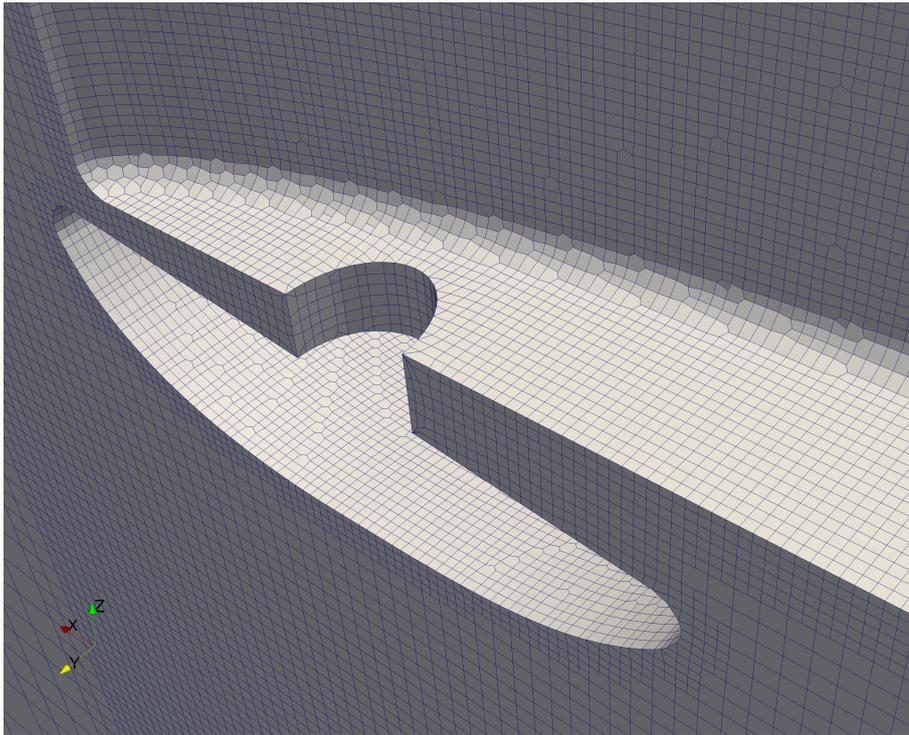


Abbildung 6.9: Ausschnitt des Netzes an der vorderen Kante

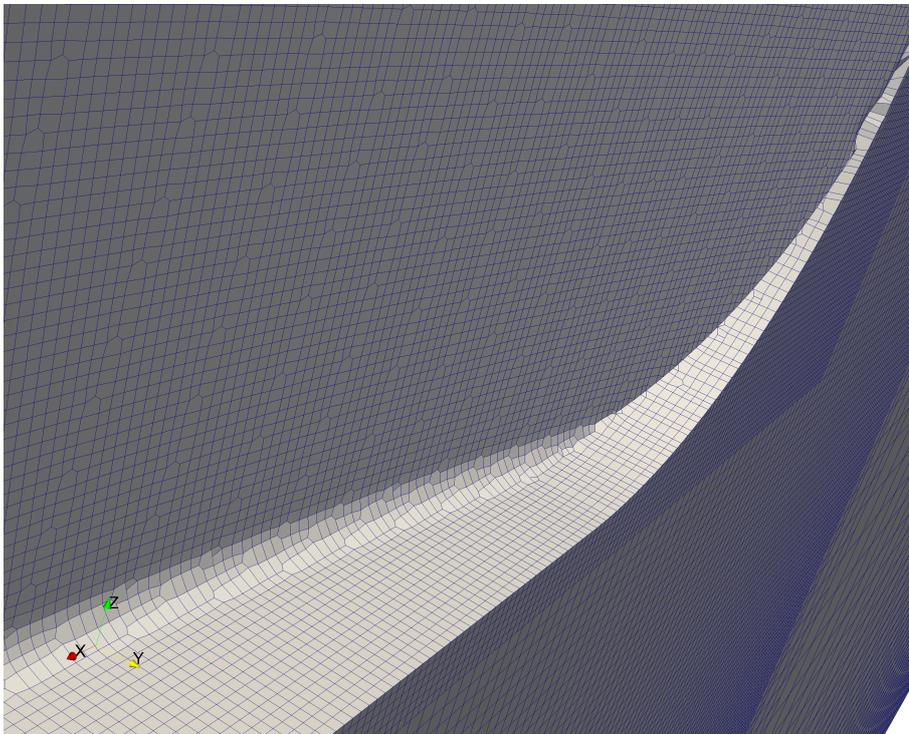


Abbildung 6.10: Ausschnitt des Netzes an der Hinterkante

Die Vernetzung des Falls ohne Halterung verläuft analog, jedoch werden die Geometrien der Halter und die nur für diese vorgesehenen Verfeinerungsregionen gelöscht. Die Richtlinien der ITTC (International Towing Tank Conference) empfehlen für Wandfunktionen, dass die erste Zelle im logarithmischen Wandbereich liegt und somit $30 < y^+ < 100$ betragen soll [9].

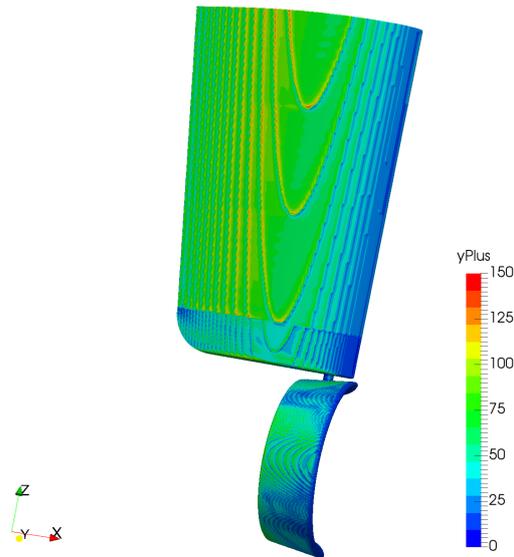


Abbildung 6.11: y^+ -Verteilung

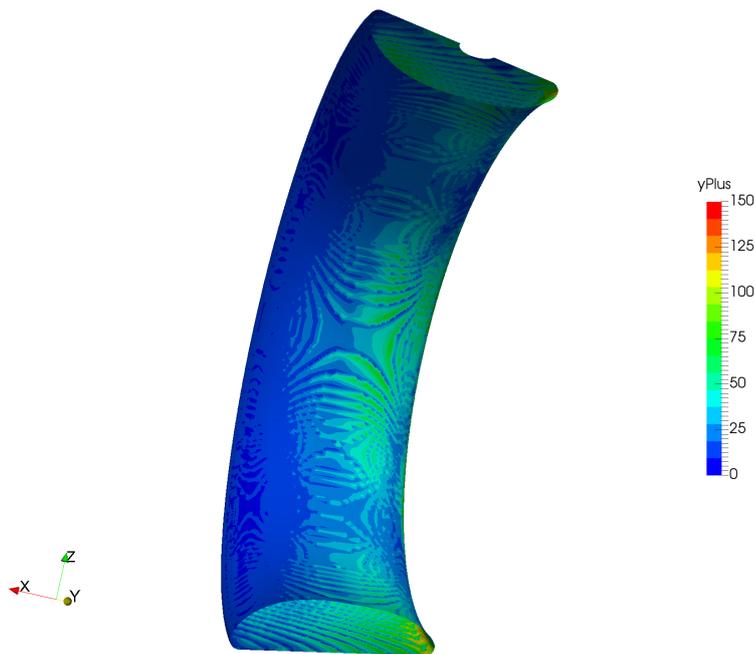


Abbildung 6.12: y^+ -Verteilung im Inneren der Düse

Durch die sehr filigrane Geometrie wurde gänzlich auf die Layer-Generierung verzichtet, da sonst die empfohlenen Werte für y^+ weit unterschritten würden. Die Abbildungen zeigen, dass die Werte in den Staupunkten sehr klein werden, was jedoch als ausreichend angenommen wurde, da hier noch keine Grenzschicht ausgebildet ist. Ansonsten bewegen sich die Werte in einem annehmbaren Bereich.

Gitterunabhängigkeitsanalyse

Um sicher zu stellen, dass die Simulationen nicht von dem numerischen Gitter abhängig sind, wird eine Gitterunabhängigkeitsanalyse durchgeführt. Hierzu wird die Zellanzahl zum einen um den Faktor $r_k = \sqrt{2}$ erhöht und zum anderen um den gleichen Faktor verringert. Ideal wäre eine Verdoppelung (Halbierung), was jedoch meist nicht praktikabel ist [10]. Dies kann ein zu grobes Netz sein oder ein so feines Netz, dass der Rechenaufwand immens wird. Hierfür wurden die Zellanzahlen im *blockMeshDict* jeweils um den Faktor r_k vergrößert/verkleinert und das Grading blieb unverändert. Auch die Verfeinerungen im *snappyHexMesh* blieben gleich, da es sich um relative Werte handelt. Abbildung 6.13 bis Abbildung 6.15 zeigen die Residuenverläufe der drei Gitter.

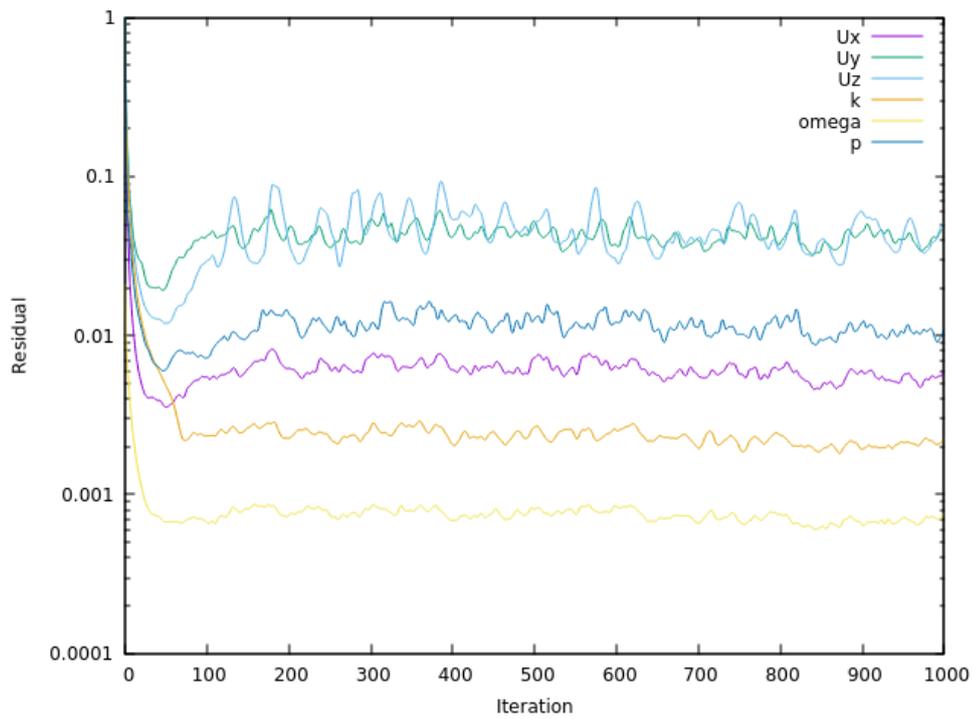


Abbildung 6.13: Residuenverlauf Gitter mit 820k Zellen

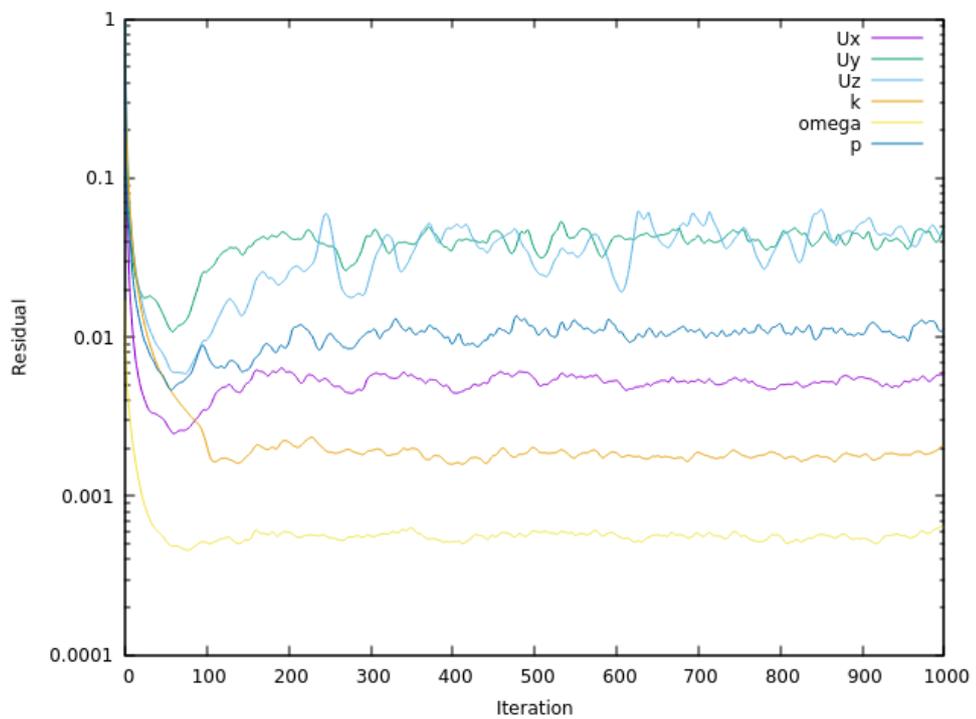


Abbildung 6.14: Residuenverlauf Gitter mit 2.1M Zellen

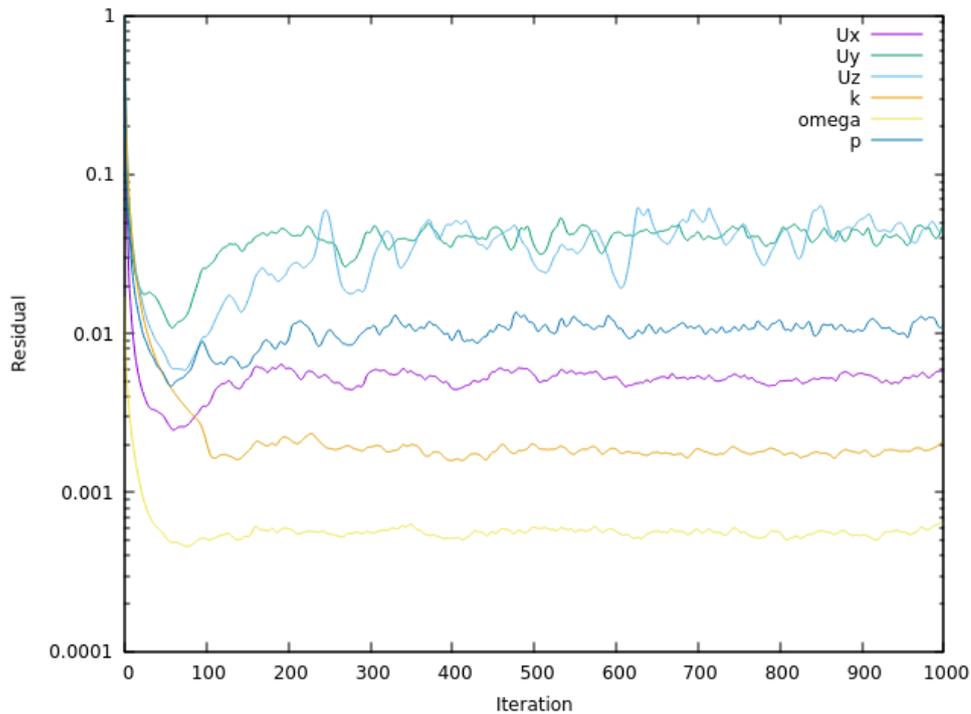


Abbildung 6.15: Residuenverlauf Gitter mit 5.6M Zellen

Die Residuen zeigen keinen wirklich befriedigenden Verlauf, die Rechnung konvergiert also vermutlich nicht, da die Residuen nicht kleiner als eine Größenordnung von 10^{-4} werden. Vergleicht man die Mittelwerte der auftretenden Kräfte auf die Düse, ist auch kein klares, unabhängiges Ergebnis zu sehen.

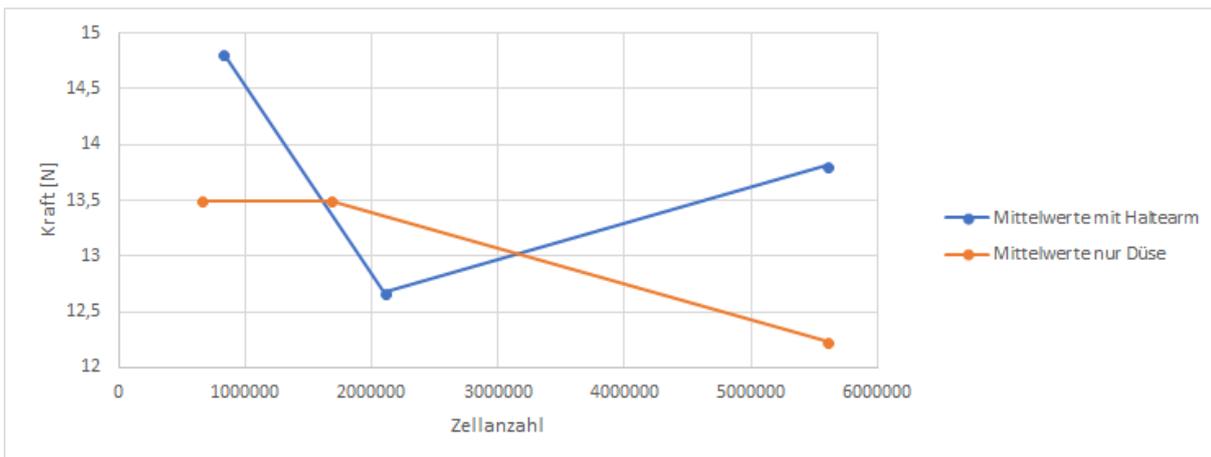


Abbildung 6.16: Kräfte zur Gitterunabhängigkeitsanalyse

Der Vergleich der Kraftverläufe zeigt ein ähnliches Bild wie die Residuen.

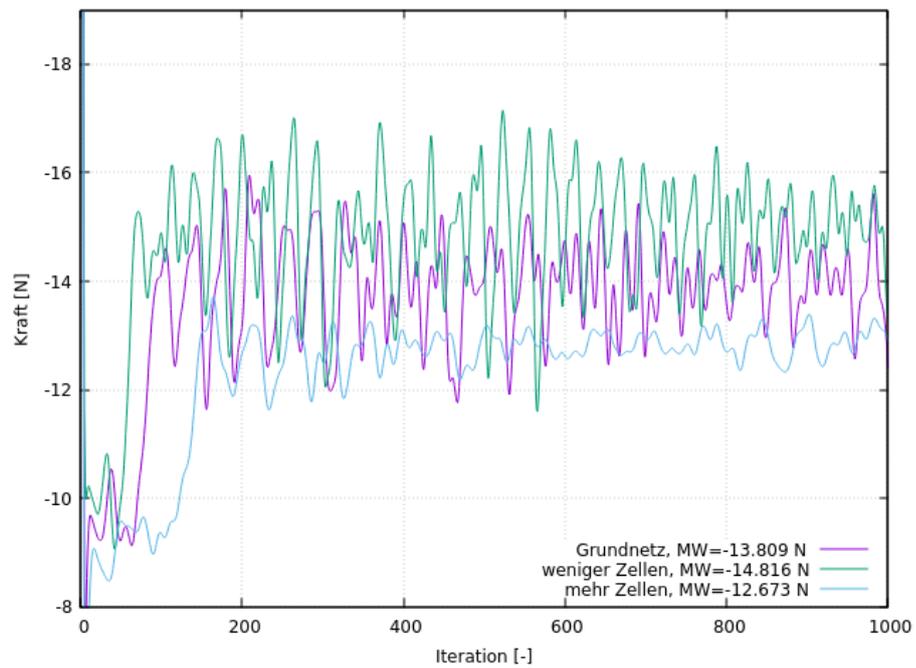


Abbildung 6.17: Kraftverlauf der Gitter mit 820k Zellen, 2.1M Zellen, 5.6M Zellen - mit Haltearm

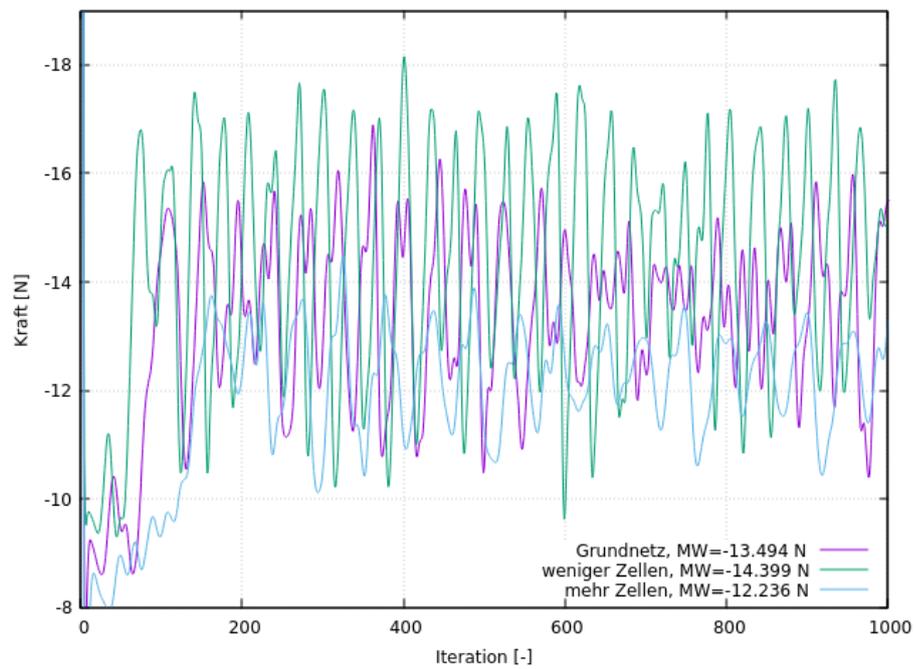


Abbildung 6.18: Kraftverlauf der Gitter mit 653k Zellen, 1.7M Zellen, 4.6M Zellen - nur Düse

6.2 Randbedingungen

Eine der wichtigsten Schritte in der CFD ist das Festlegen der Randbedingungen. Der generelle Aufbau der angestrebten Simulation sieht folgendermaßen aus:

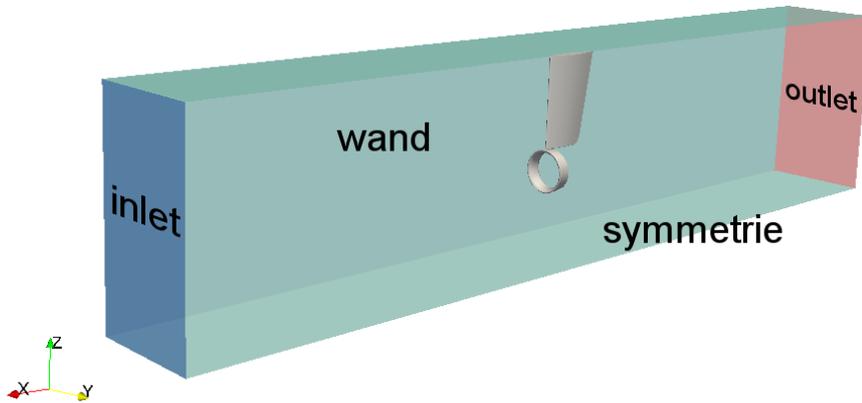


Abbildung 6.19: Schema Berechnungsgebiet

Durch das *inlet* (blau) soll das Fluid einströmen. Das Berechnungsgebiet wird einerseits durch die *wand* (grün) und die Symmetrie-Ebene *symmetrie* begrenzt. Ausströmen soll das Fluid am *outlet* (rot). Diese Werte werden im **Ordner 0** mit den Dateien *U*, *p*, *k*, *omega* und *nut* festgelegt. Für jede Begrenzung der Domain müssen also Zustände zum Zeitpunkt 0 vorgegeben werden. Eine Übersicht zeigen Tabelle 6.3 und Tabelle 6.4.

	Geschwindigkeit \mathbf{U}	Druck \mathbf{p}	turbulente Scheinviskosität ν_t
internalField	-2 m/s	$0 \text{ m}^2/\text{s}^2$	$0 \text{ m}^2/\text{s}$
inlet	fixedValue ($= -2 \text{ m/s}$)	zeroGradient	calculated
outlet	inletOutlet	fixedValue	calculated
symmetrie	symmetry	symmetry	symmetry
halter	noSlip	zeroGradient	nutkWallFunction
schwert	noSlip	zeroGradient	nutkWallFunction
duese	noSlip	zeroGradient	nutkWallFunction
wand	symmetry	symmetry	symmetry

Tabelle 6.3: Randbedingungen für \mathbf{U} , \mathbf{p} und ν_t

Das *internalField* beschreibt den Zustand des Berechnungsgebietes, bevor die Simulation in Kraft tritt. Für die Geschwindigkeit wird sowohl für das *internalField* als auch für das *inlet* eine Geschwindigkeit von 2 m/s in x -Richtung angenommen. Da es sich um ein inkompressibles Fluid handelt, wird als Anfangswert ein Druck von $0 \text{ m}^2/\text{s}^2$ angenommen, weil nur die Druckänderung eine Rolle spielt und so Differenzen leicht zu erkennen sind. Die Einheit des Drucks ergibt sich aus der Eigenschaft, dass der Druck mit der Dichte normiert wird, also $p_{\text{normiert}} = \frac{p}{\rho}$ mit $[p] = \text{m}^2/\text{s}^2$. Am *outlet* soll sich ein **zeroGradient**

	turb. kin. Energie k	Turbulenzfrequenz ω
internalField	0.06 m ² /s ²	24.5 1/s
inlet	fixedValue(= 0.001 35 m ² /s ²)	fixedValue(= 3.67 1/s)
outlet	inletOutlet	inletOutlet
symmetrie	symmetry	symmetry
halter	kqRWallfunction	omegaWallFunction
schwert	kqRWallfunction	omegaWallFunction
duese	kqRWallfunction	omegaWallFunction
wand	symmetry	symmetry

Tabelle 6.4: Randbedingungen für k und ω

für den Druck einstellen. An den zu untersuchenden Geometrien werden die `noSlip` Bedingungen eingestellt. Da die Funktionen bereits in Abschnitt 5.3 erklärt wurden, soll hier nicht weiter auf jede Einstellung eingegangen werden. Die Initialwerte von k und ω werden nach folgenden Zusammenhängen und Richtwerten berechnet.

Die turbulente kinetische Energie k wird mit

$$k = 1,5(U \cdot Tu)^2 \quad (6.1)$$

berechnet, wobei Tu die Turbulenzintensität ist. Aus Erfahrungswerten der CFD-Abteilung der HSVA wurden die Turbulenzintensitäten für den großen Schlepptank übernommen. Dabei beträgt die Turbulenzintensität in der Domain 10% und am Einlass 1,5%. Die tatsächlichen Werte im Schlepptank sind als geringer einzuschätzen und die genutzten Werte sind als numerische Randbedingungen anzusehen.

Mit dem Zusammenhang

$$\omega = \frac{\sqrt{k}}{L_T} \quad (6.2)$$

kann auch ω berechnet werden. Hierbei ist L_T das turbulente Längenmaß, das hier mit 0.01 m abgeschätzt wurde, da auftretende Wirbel als kleiner geschätzt werden.

6.3 Simulationseinstellungen

Gleichungslöser

In der Datei `fvSolution` befinden sich die Angaben zu den Lösern für die linearen Gleichungssysteme.

Variable	OpenFoam-Name	Solver
Druck	p	GAMG
Geschwindigkeit	U	smoothSolver
turbulente kin. Energie	k	smoothSolver
Turbulenzfrequenz	omega	smoothSolver

Tabelle 6.5: Solver Einstellungen aus *fvSolution*

Der GAMG-Solver (Geometric agglomerated algebraic multigrid solver) fasst bei der Berechnung einige Zellen zusammen und teilt diese danach wieder auf die einzelnen Zellen auf. Dadurch ist der GAMG-Solver verhältnismäßig schnell. Für die *smoother* wurde die Einstellung `GaussSeidel` verwendet. Eine Übersicht zeigt Tabelle 6.5. Diese Einstellungen wurden aus den Tutorials der OpenFoam-Installation wie z.B. *pitzDaily* oder *motorBike* übernommen und sollen an dieser Stelle nicht weiter erklärt werden. Mit der Einstellung

```
SIMPLE
{
nNonOrthogonalCorrectors 0;
consistent yes;
}
```

wird das SIMPLEC-Verfahren eingeschaltet.

Rechnungssteuerung

Die Angaben, wie die Rechnung in Bezug auf die Programmfunktionen verlaufen soll, werden in der Datei *controlDict* gemacht. Hier wird bestimmt, mit welchem Lösungsalgorithmus gerechnet werden soll. Für den SIMPLE-Algorithmus wird

```
application simpleFoam;
```

angegeben. Außerdem werden hier die Zeitschritte oder die Rechendauer eingestellt. Da bei einer stationären Rechnung der Zeitschritt nur den Iterationsschritt angibt, wurde dieser auf den Wert 1 gesetzt. Am Ende der Datei können noch Angaben über Funktionen gemacht werden, die der Aufbereitung der Daten dienen. In diesem Fall wurden die Funktionen für Kräfte und für den Wert von y^+ genutzt. Die Funktionen für die Kräfte wurden in ausgegliederten Dateien folgendermaßen (auszugsweise) spezifiziert.

```
patches ( "duese" );
rho rhoInf;
log true;
rhoInf 1000;
porosity no;
CofR (0 0 0);
pitchAxis (0 1 0);
```

Hier wird angegeben, dass die Kraft auf der Fläche der Düse berechnet werden soll. Die Dichte wird über `rhoInf` angegeben, was jedoch bei inkompressiblen Fluiden redundant ist. Außerdem wird noch mit `CofR` und `pitchAxis`, also dem Mittelpunkt der Rotation und der Drehachse, ein Drehpunkt für die Momentenberechnung angegeben, welche in diesem Fall nicht von Interesse war.

Fluid- und Turbulenzeigenschaften

Im **constant Ordner** befinden sich einerseits der STL Dateien im Ordner *triMesh* und die Netzinformationen im Ordner *polyMesh*. In der Datei *turbulenceProperties* wird das Turbulenzmodell und das RANS-Verfahren (in OpenFOAM RAS) festgelegt.

```
simulationType RAS;
RAS
{
  RASModel kOmegaSST;
  turbulence on;
  printCoeffs on;
}
```

In der Datei *transportProperties* wird die Art des Fluids als newtonsches Fluid eingestellt und die kinematische Viskosität ν mit $1 \cdot 10^{-6} \frac{m^2}{s}$ angegeben.

6.4 Implementierung eines Propellers

Die Implementierung eines Propellers sollte mit der Funktion *actuatorDisk* von OpenFOAM realisiert werden. Diese implementiert die Strahltheorie nach Froude/Rankine. Dazu wird zunächst eine Geometrie angelegt, in der der Drucksprung aufgebracht werden soll. Zu diesem Zweck stehen als Auswahl die gängigen geometrischen Formen wie ein Kreis oder ein Rechteck zur Verfügung. Für den Propeller wurde also eine Kreisscheibe mit dem Radius von 0.1075 m und einer Dicke von 0.02 m im Koordinatenursprung erzeugt. Diese Einstellungen werden in einer Datei namens *topoSetDict* im Ordner **system** angelegt und mit dem Befehl `topoSet` ausgeführt. Dies legt ein *cellSet* an, die Kreisscheibe wird nur auf ganze Zellen angewendet. In einer weiteren Datei im Ordner **constant** namens *fvOptions* werden die Parameter für das *cellSet* angegeben. Hier werden die Funktion *actuatorDisk*, deren Richtung, Fläche und Leistungseigenschaften festgelegt. Diese sind der Leistungskoeffizient c_P und der Schubkoeffizient c_T . Da dies Werte sind, die in der Propellertheorie geläufig sind, wurden diese aus einem Beispiel-Propeller der HSVA verwendet. Dies führte zu Ergebnissen, die die Propellerwirkung negativ abbildeten. Eine Drehung der Scheibe führte zu Konvergenzproblemen und viel zu hohen Geschwindigkeiten. Die Betrachtung des Quellcodes ergab, dass die *actuatorDiskSource* die Koeffizienten folgendermaßen berechnet:

$$a = 1 - \frac{c_P}{c_T} \quad (6.3)$$

$$T = 2\rho AU_0^2 a(1 - a) \quad (6.4)$$

$$U_1 = (1 - a)U_0 \quad (6.5)$$

wobei A die Fläche der Scheibe ist, U_0 die Geschwindigkeit der ungestörten Zuströmung und U_1 die Geschwindigkeit an der Scheibe.

Berechnet man mit diesen Gleichungen den Schub eines Schiffspropellers, stimmt dieser nicht mit dem tatsächlichen Schub überein. Ein Versuch, die Koeffizienten über den Schub zu berechnen, ergab erneut Konvergenzprobleme. Nach weiterer Recherche wurde klar, dass diese Funktion nur auf Windkraftanlagen anwendbar ist. Die Berechnung beider Fälle basiert zwar auf der gleichen Theorie, jedoch wird hier mit anderen Vorzeichen gearbeitet, da die Leistung bei Windenergieanlagen aus der Strömung entnommen wird, während bei Schiffspropellern die Leistung durch den Propeller aufgebracht wird.

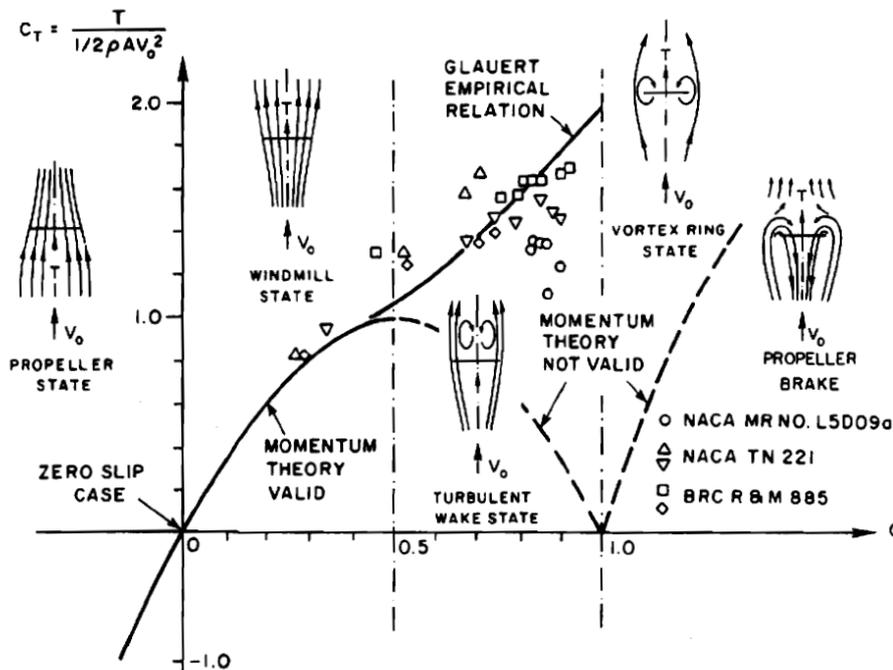


Abbildung 6.20: Diagramm Schubbeiwert c_T über den axialen Induktionsfaktor a [3]

In Abbildung 6.20 sieht man die Einordnung des Faktors a . Gängige Windenergieanlagen werden in einem Bereich von $0 < a < 0,5$ betrieben. Interessant ist für diese Arbeit der Bereich links von y-Achse, der mit *Propeller State* beschrieben wird. In diesem Bereich, wo a negativ wird, werden also Propeller betrieben. Für diesen Fall ist jedoch die *actuatorDisk* nicht vorgesehen.

An dieser Stelle soll auf eine Herleitung des Induktionsfaktors und die Erklärung der Strahltheorie nach Froude/Rankine verzichtet werden, da es für das Nachvollziehen dieser

Arbeit keine Relevanz hat. Für weitere Informationen sei Kapitel 5 in [11] empfohlen, da es sowohl die Herleitung für den Propeller als auch für die Windturbine kurz behandelt. Eine weitere Herangehensweise ist die Implementierung des Drucksprungs über eine *fixed-Jump* Randbedingung. Hierzu wurden bereits Berechnungen durchgeführt, die vielversprechende Ergebnisse liefern konnten. Es kam bei der Implementierung jedoch zu Problemen mit dem Vernetzen und der Berechnung auf mehreren Prozessoren. Aus Zeitgründen konnte diese Variante nicht weiter verfolgt werden. [15] beschreibt die Verfahrensweise, um den Propeller zu implementieren, sehr genau und soll hier als Hinweis genügen. Aus genannten Gründen musste darauf verzichtet werden, die Düse mit implementiertem Propeller zu modellieren und zu simulieren.

7 Vergleich der Ergebnisse

Es handelt sich in diesem Kapitel, wenn nicht anders angegeben, um Ansichten des letzten Iterationsschrittes von 1000.

Um die Ergebnisse möglichst übersichtlich darzustellen, wurden in diesem Kapitel die Texte auf der linken Seite und die dazugehörigen Abbildungen auf der rechten Seite platziert.

7.1 Geschwindigkeitsverläufe

Die Gesamtansichten aus der Symmetrie-Ebene zeigen bereits einen Überblick über den Geschwindigkeitsverlauf. Die Geschwindigkeit verringert sich im Bereich hinter dem Schwert. Auffällig ist bei Abbildung 7.2, dass die Geschwindigkeiten nicht symmetrisch sind.

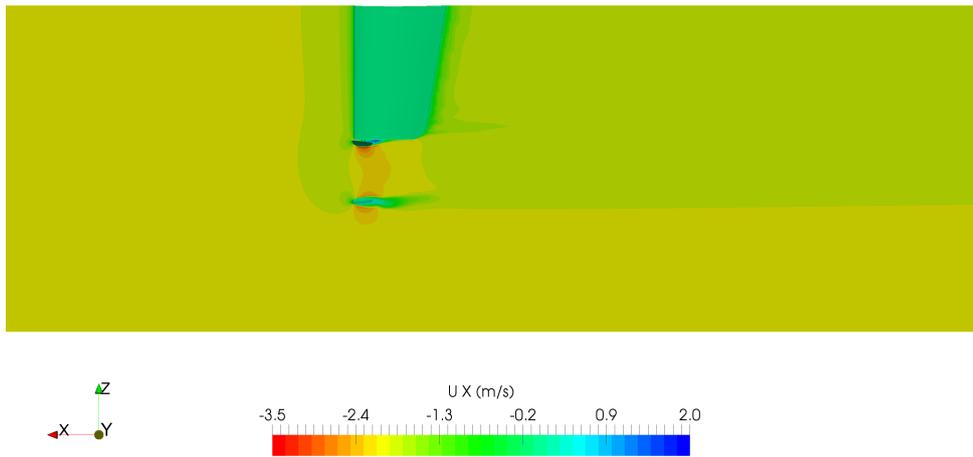


Abbildung 7.1: Geschwindigkeit in x-Richtung - Gesamtansicht - mit Halter

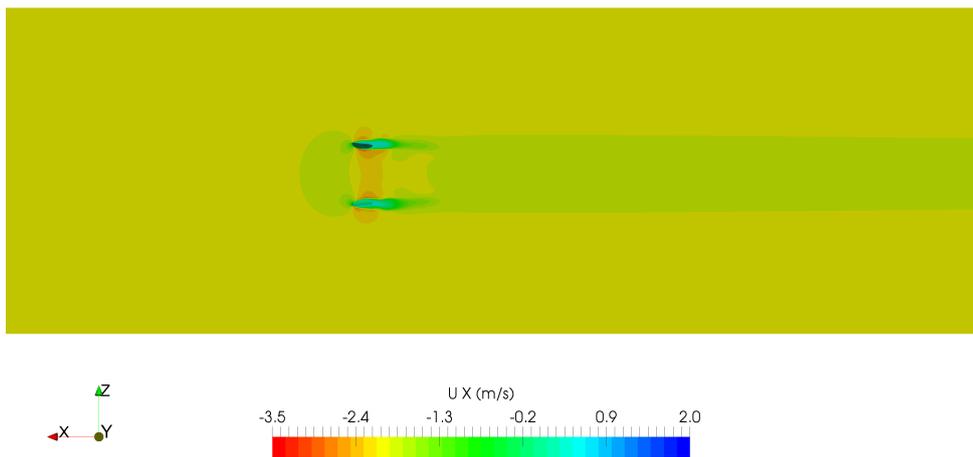


Abbildung 7.2: Geschwindigkeit in x-Richtung - Gesamtansicht - ohne Halter

Abbildung 7.3 und Abbildung 7.4 zeigen den Bereich der Düse etwas genauer. Man erkennt, dass bei beiden Fällen zwar die Geschwindigkeit hinter den Bauteilen abnimmt, aber besonders auffällig ist hier der stark verzögerte Bereich hinter dem Teil der Düse, der am Haltearm befestigt ist. Dies war zu erwarten, da der Haltearm natürlich einen Widerstand gegen die Strömung darstellt. In beiden Abbildungen erkennt man auch die Beschleunigung der Strömung innerhalb der Düse. Diese ist in Abbildung 7.3 stärker ausgeprägt. Auch hier wird die Asymmetrie in Abbildung 7.4 deutlich. Diese ist untypisch, da es sich um ein komplett symmetrisches Problem handelt. Einerseits durch die nicht konvergierenden Residuenverläufe und andererseits durch die erkennbare Asymmetrie wird die Vermutung angestellt, dass es sich nicht um eine stationäre Lösung handelt. Abbildung B.2 bis Abbildung B.7 zeigen verschiedene Iterationsschritte der Rechnung. Es sind unterschiedliche Geschwindigkeitsverläufe zu sehen, was gegen eine stationäre Lösung spricht. Zu beachten ist, dass es sich hier um Iterationsschritte handelt und kein physikalisch sinnvoller, zeitlicher Verlauf zu sehen ist.

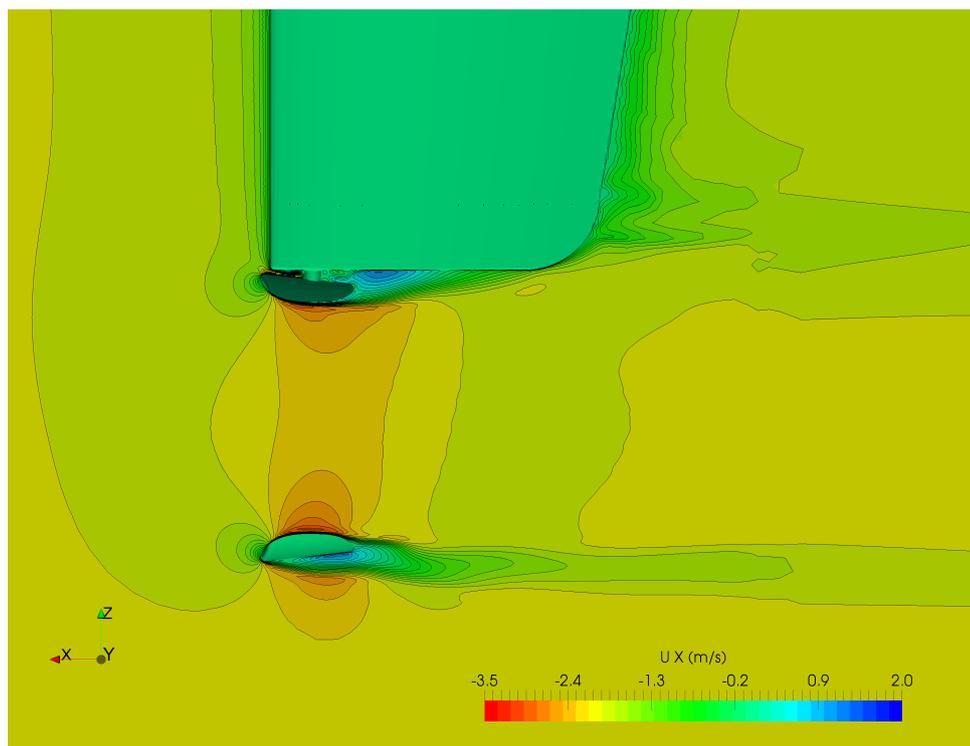


Abbildung 7.3: Geschwindigkeit in x-Richtung - Detailansicht - mit Halter

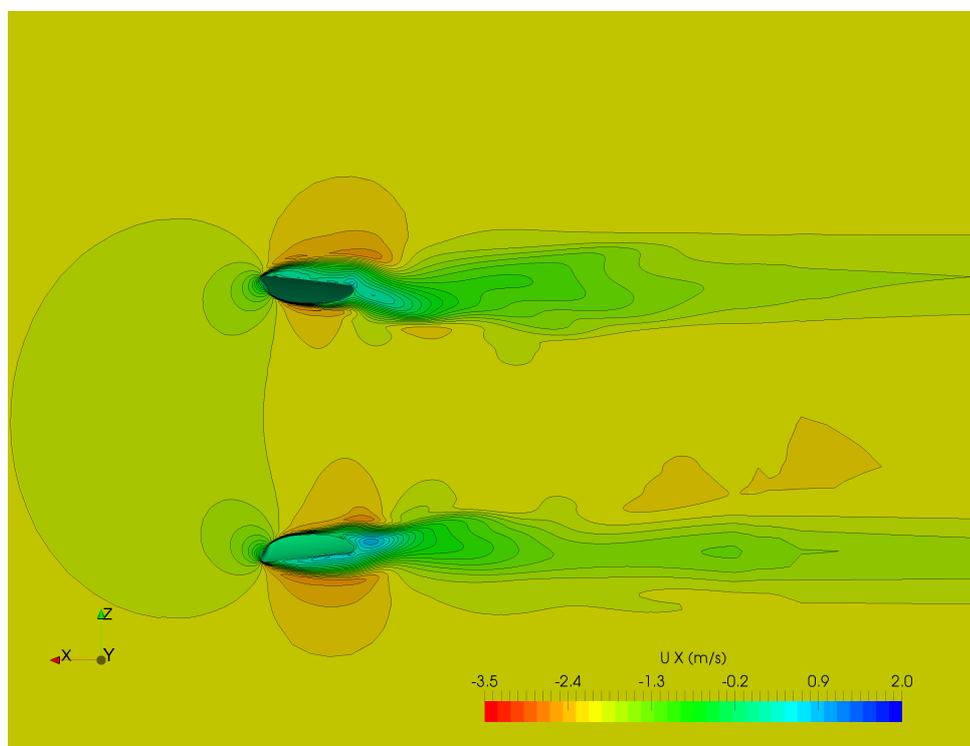


Abbildung 7.4: Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter

Die Stromlinien zeigen den Verlauf der umströmenden Partikel und wie sie um die Düse herum verlaufen. In dieser perspektivischen Darstellung wird das Zusammenspiel von Geometrie und Strömung gut deutlich. Besonders die Wirbelbildung ist erwähnenswert. Die mehrfach erwähnte Asymmetrie wird auch bei diesen Abbildungen wieder deutlich. Diese und die auffällige Ablösung der Strömung, die zur Wirbelbildung führt, sprechen dafür, dass die Lösung instationär ist. Die Wirbel verursachen eine Rückströmung. Diese ist vor allem am Bereich des Übergangs zwischen Düse und Haltearms stärker ausgeprägt. Dies ist besonders interessant, da eine Rückströmung von hinten eine Kraft entgegen der Hauptströmungsrichtung auf die Düse verursacht. Dadurch können die Kraftmessungen eine geringere Kraft ergeben.

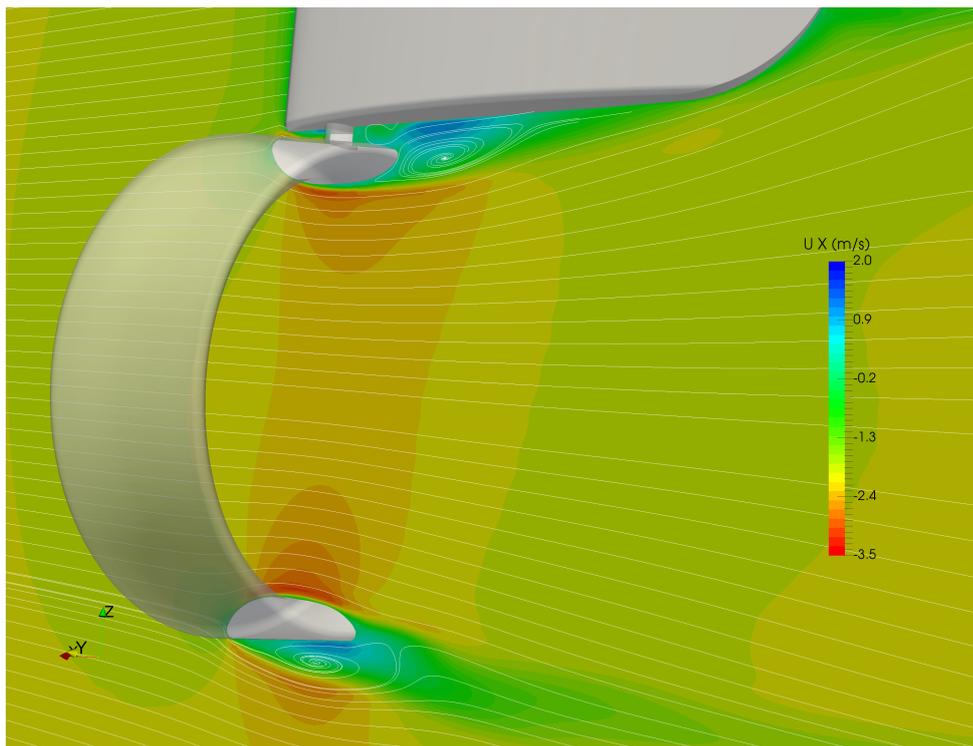


Abbildung 7.5: Geschwindigkeit in x-Richtung - Stromlinien perspektivisch - mit Halter

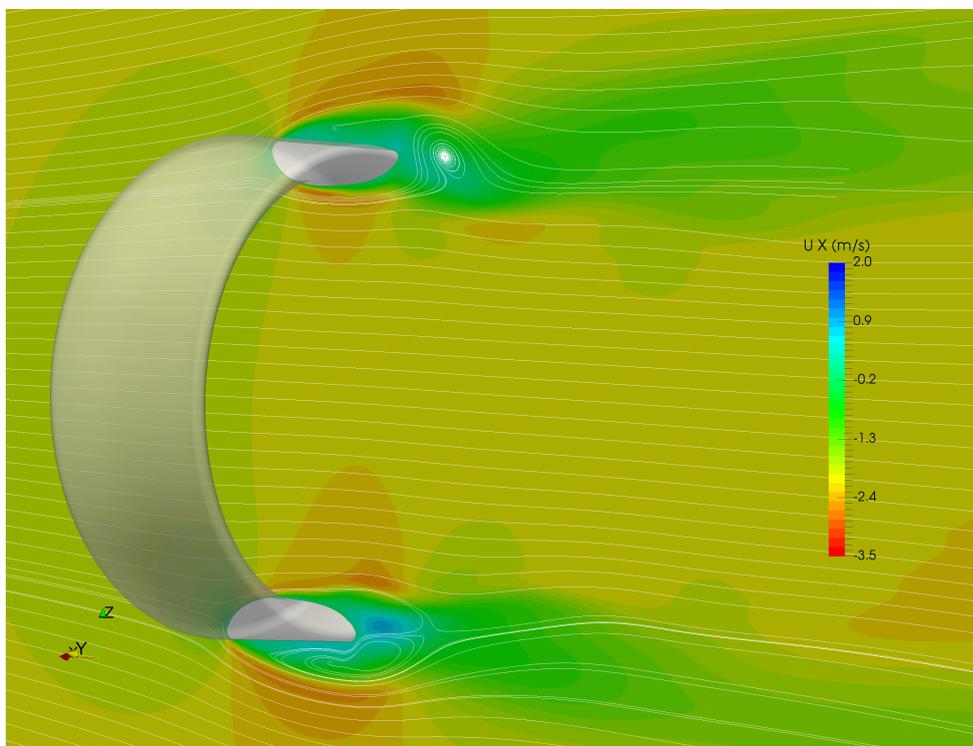


Abbildung 7.6: Geschwindigkeit in x-Richtung - Stromlinien perspektivisch - ohne Halter

In der Vektor-Darstellung sieht man die Wirbelansätze im Bereich der Düse erneut. Hier erkennt man zusätzlich, dass die Hinterkante des Schwertes die Strömung nach oben hin ablenkt. Dies wird für diese Anwendung als wenig relevant angesehen, da dies nur Auswirkungen auf den Nachstrom mit sich bringt. Würde in diesem Bereich jedoch ein messtechnisch relevantes Bauteil angebracht werden, würde es an Bedeutung gewinnen.

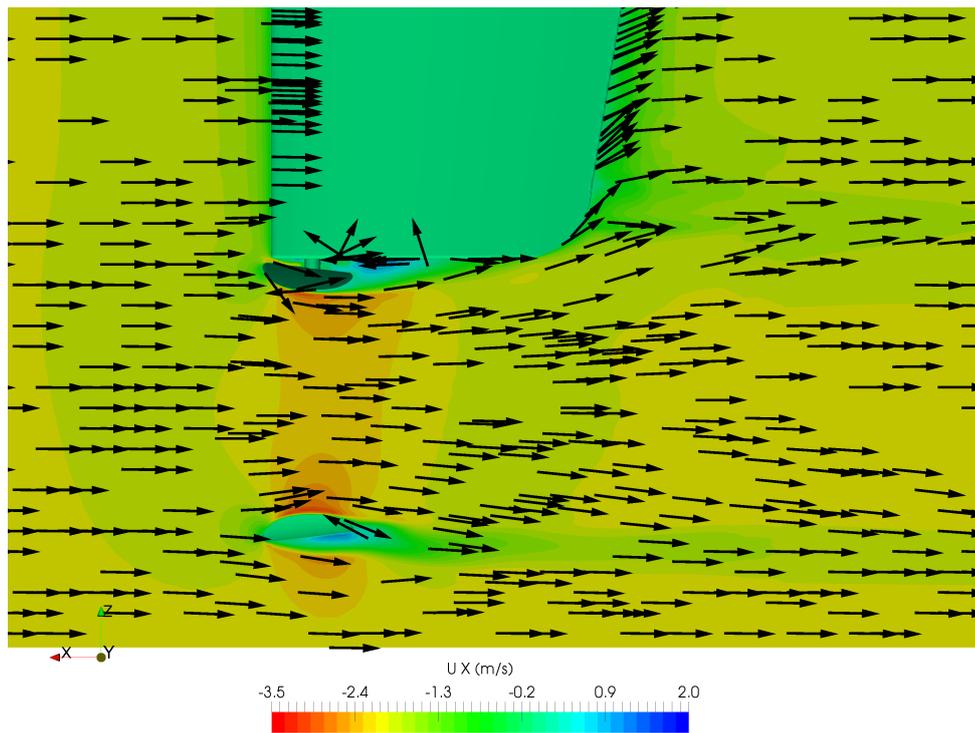


Abbildung 7.7: Geschwindigkeit in x-Richtung - Vektorpfeile - mit Halter

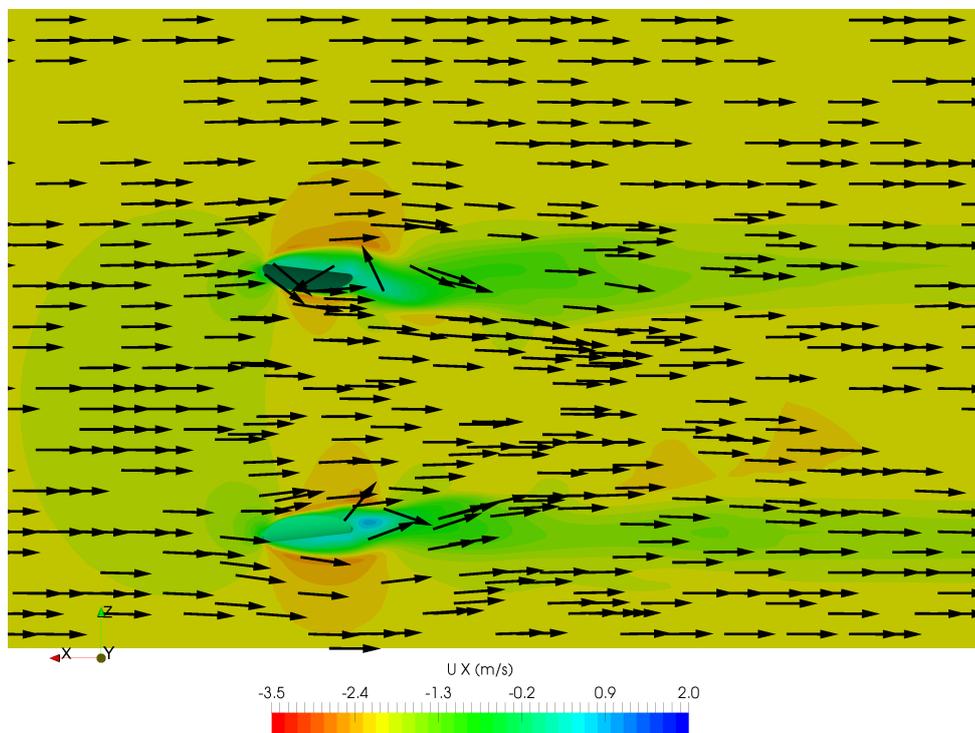


Abbildung 7.8: Geschwindigkeit in x-Richtung - Vektorpfeile - ohne Halter

Bei der Darstellung der Stromlinien im dreidimensionalen Raum ist auch die Umlenkung durch die Hinterkante des Schwertes zu sehen. Im Fall der Düse allein wird die Strömung lediglich umgelenkt und findet den Weg zurück zur ursprünglichen Bahn. Besonders sind die verwirbelten Bereiche hervorzuheben, die auf den vorigen Abbildungen bereits sichtbar waren. Es wird deutlich, dass die Partikel keine gerade Bahn in x-Richtung durchlaufen, sondern durch die Wirbel in jede Raumrichtung abgelenkt werden.

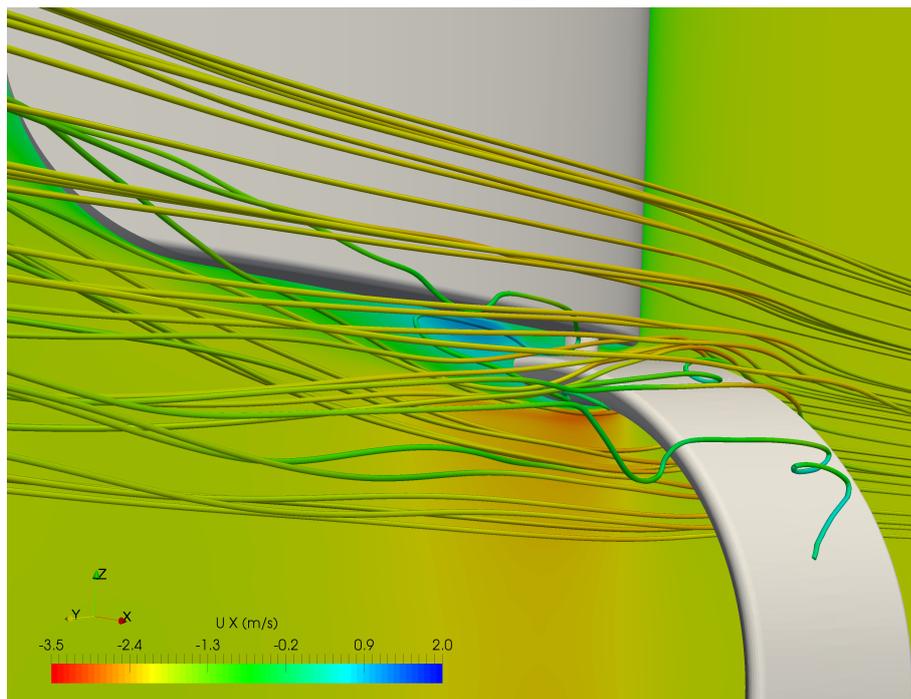


Abbildung 7.9: Stromlinien im Raum - mit Halter

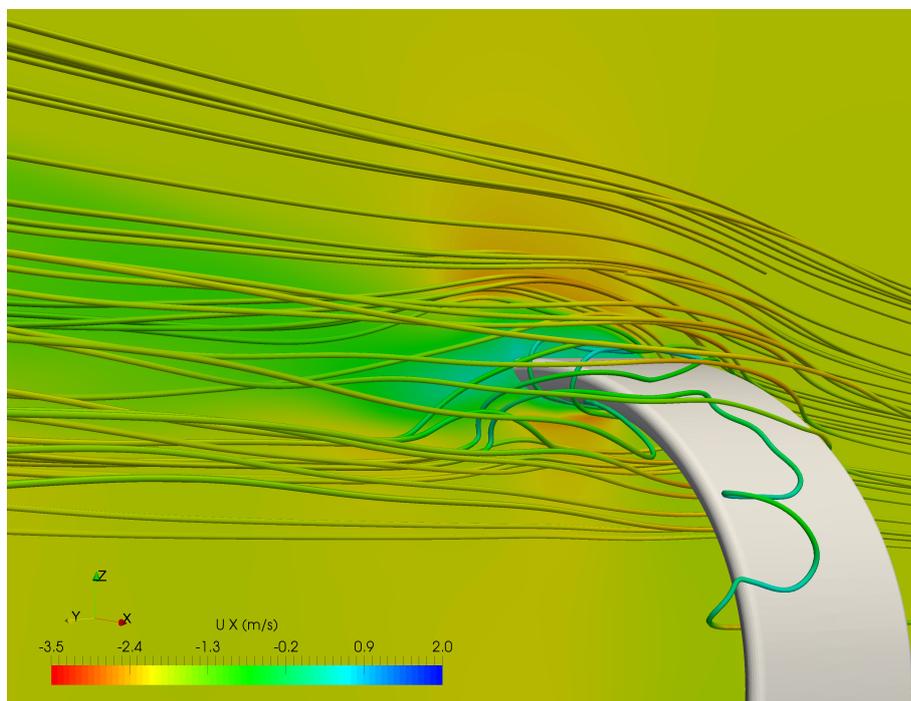


Abbildung 7.10: Stromlinien im Raum - ohne Halter

Diese Verwirbelungen werden auch im Schnitt durch die xy -Ebene deutlich. Trotz des symmetrischen Aufbaus unterscheiden sich die Schnittebenen. Die Strömung um die symmetrischen Körper herum sind im Mittel vermutlich symmetrisch, was jedoch die stationäre Rechnung nicht auflösen kann.

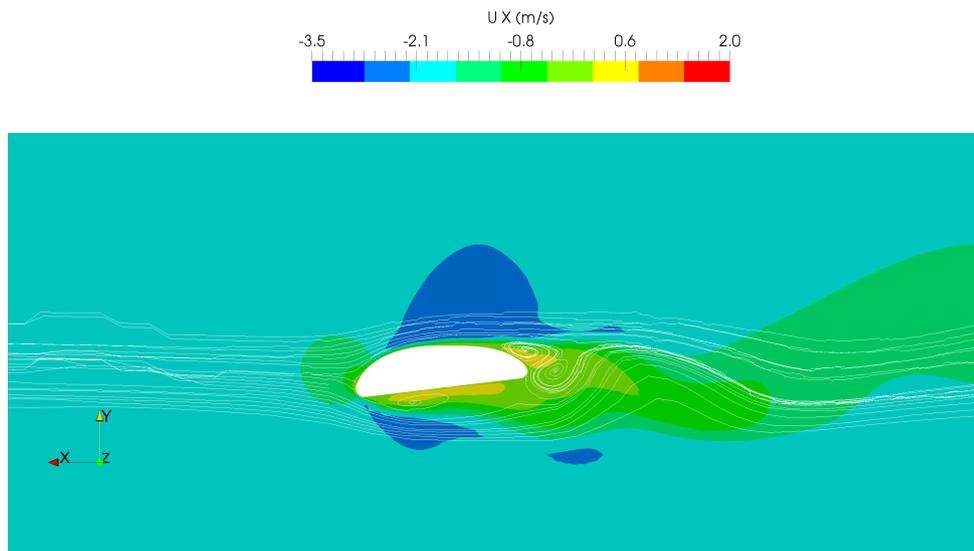


Abbildung 7.11: Stromlinien in der xy-Ebene - mit Halter

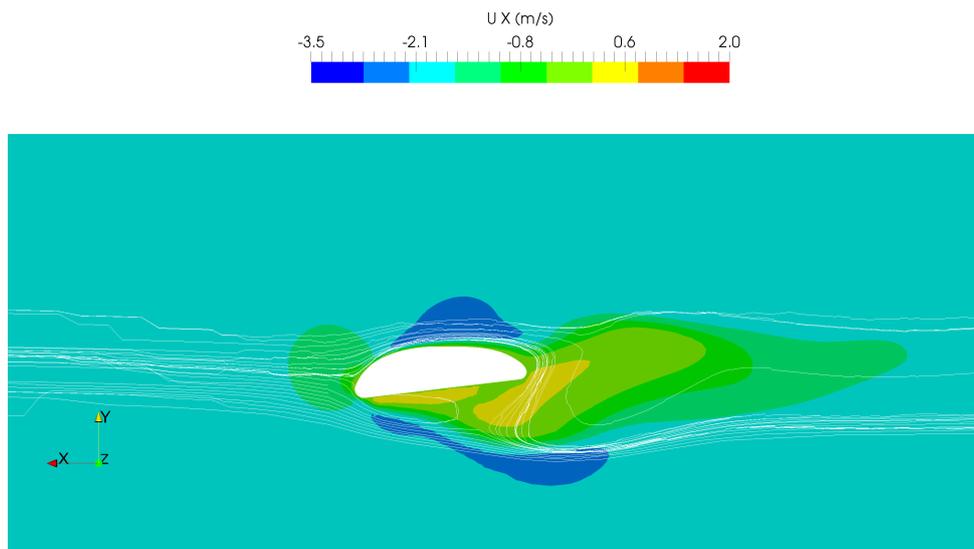


Abbildung 7.12: Stromlinien in der xy-Ebene - ohne Halter

Eine Darstellung des Halters im Schnitt mit Stromlinien zeigt, dass der Haltearm eine strömungsgünstige Form aufweist. Naturgemäß stellt der Haltearm eine Störung dar, jedoch wird die Strömung hier ohne Unregelmäßigkeiten umgelenkt und es zeigen sich keine Ablösungen. Hier wurde die Skala der Geschwindigkeit bewusst verändert, um den Bereich an der Wand zu verdeutlichen, an dem die Geschwindigkeit zu null wird. Da in dieser Ansicht keine Bereiche mit einer Rückströmung zu sehen sind, ist dies hier besonders gut sichtbar.

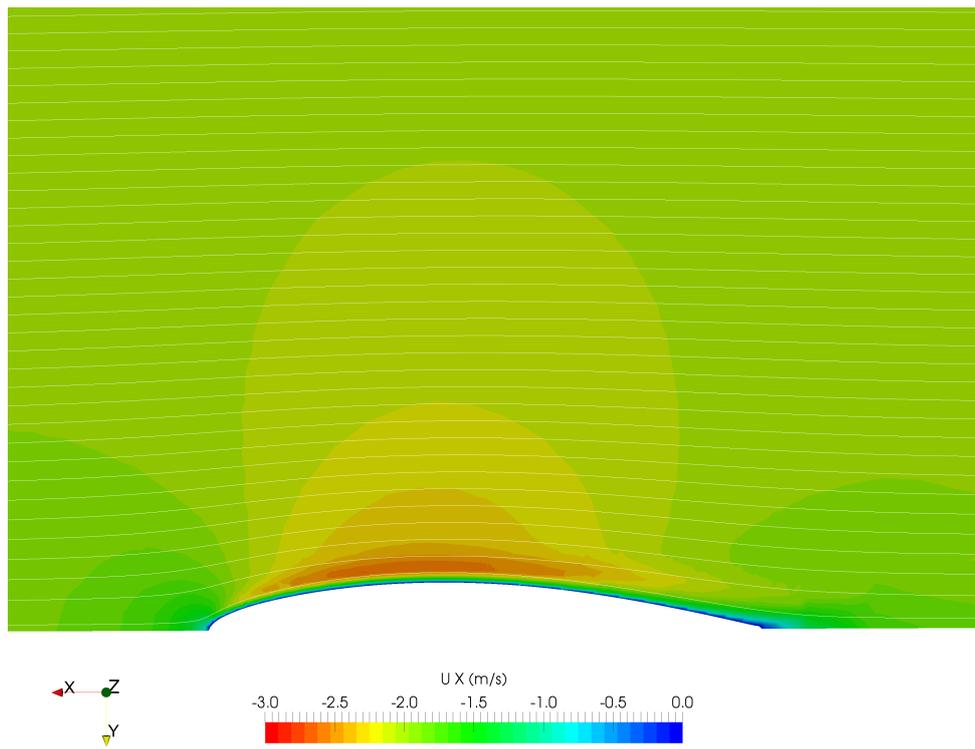


Abbildung 7.13: Stromlinien um den Haltearm

7.2 Druckverteilung

Im Staupunkt des Haltearms ist die Druckerhöhung zu sehen. Auch in Abbildung 7.15 sieht man den Effekt eines Staupunkts vor der Stirnseite der Düse. An Innenseite der Düse erkennt man einen Bereich mit einem geringeren Druck. Hier wird deutlich, wieso die Strömung innerhalb der Düse (zu sehen in Abbildung 7.3 ff.) beschleunigt wird. Der Druck fällt durch die Form der Düse im Inneren ab. Die Düse ist ähnlich wie ein Tragflügelprofil geformt und dadurch ergeben sich eine Druck- und eine Saugseite. Zu erkennen ist dies am Unterdruck im Inneren der Düse und am leichten Überdruck an der Außenseite. Dieses Verhalten kann durch die Bernoulli-Gleichung erklärt werden.

Die allgemeine Bernoulli-Gleichung lautet

$$p_t = p + \rho g z + \frac{\rho}{2} u^2 = \text{const} \quad (7.1)$$

mit der Geschwindigkeit u , dem Totaldruck p_t , dem Betriebsdruck p , der Dichte ρ , der Höhe z und der Erdbeschleunigung g .

Diese Gleichung sagt aus, dass, wenn der Druck abnimmt, die Geschwindigkeit zunehmen muss (oder umgekehrt).

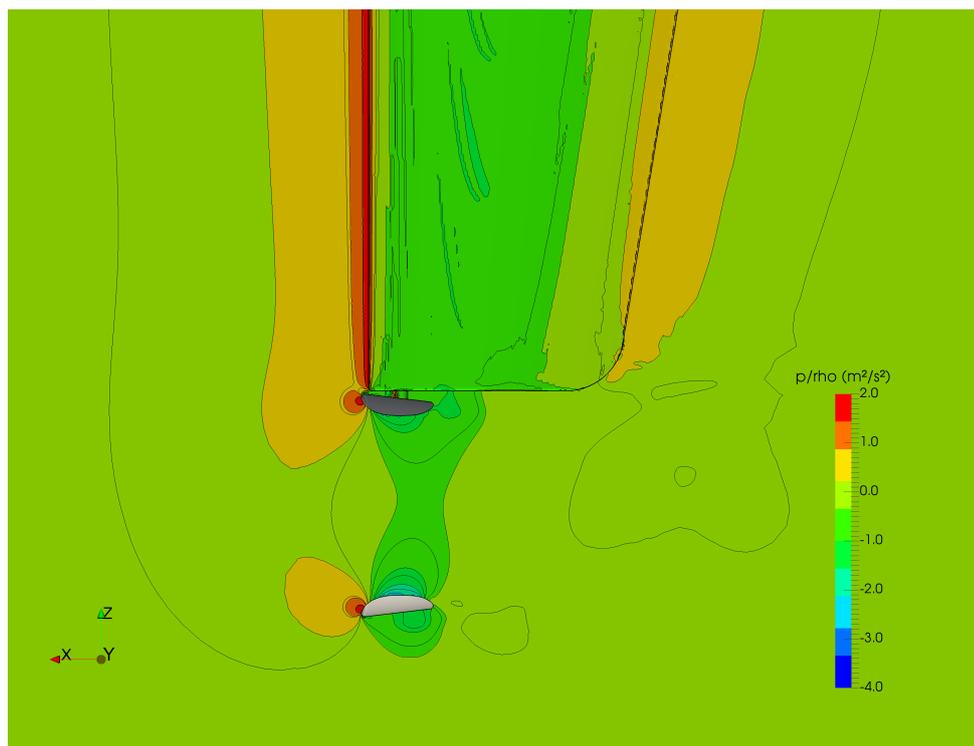


Abbildung 7.14: Druck - mit Halter

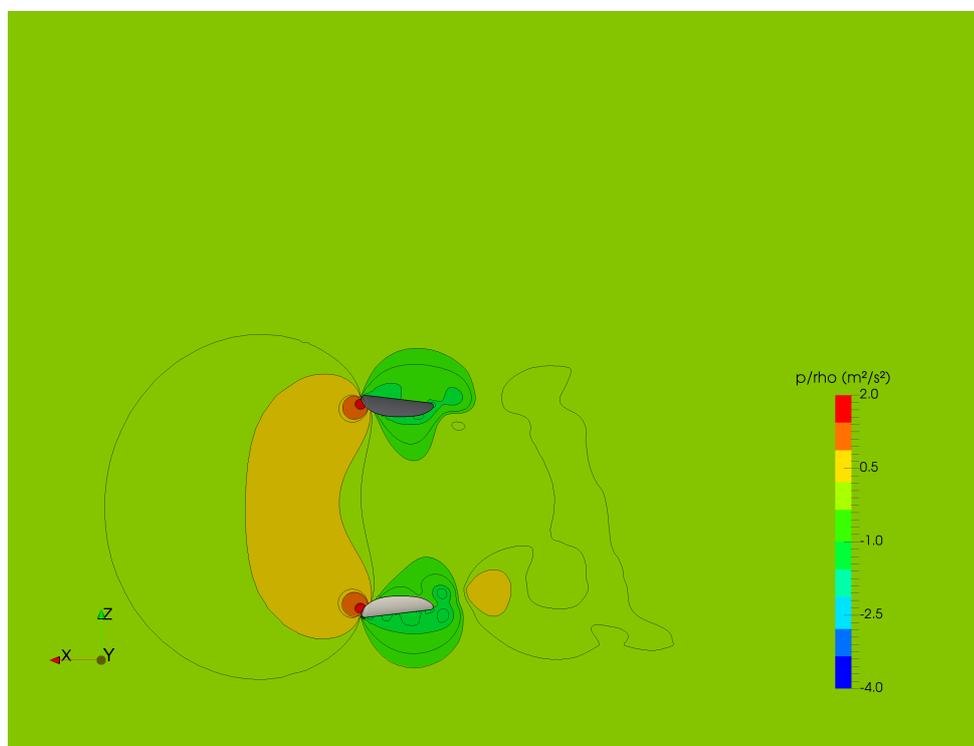


Abbildung 7.15: Druck - ohne Halter

Stellt man den Druck auf der Oberfläche dar, sieht man, dass in den Staupunkten der Druck stark erhöht ist. An den Seiten fällt der Druck auf einen leicht negativen Wert. Hier wird die Strömung also etwas beschleunigt. Abbildung B.8 bis Abbildung B.11 zeigen die Ansicht in verschiedenen Iterationsschritten. Auch hier wird deutlich, dass die Rechnung vermutlich instationär ist, weil sich die Druckausbildungen unterscheiden.

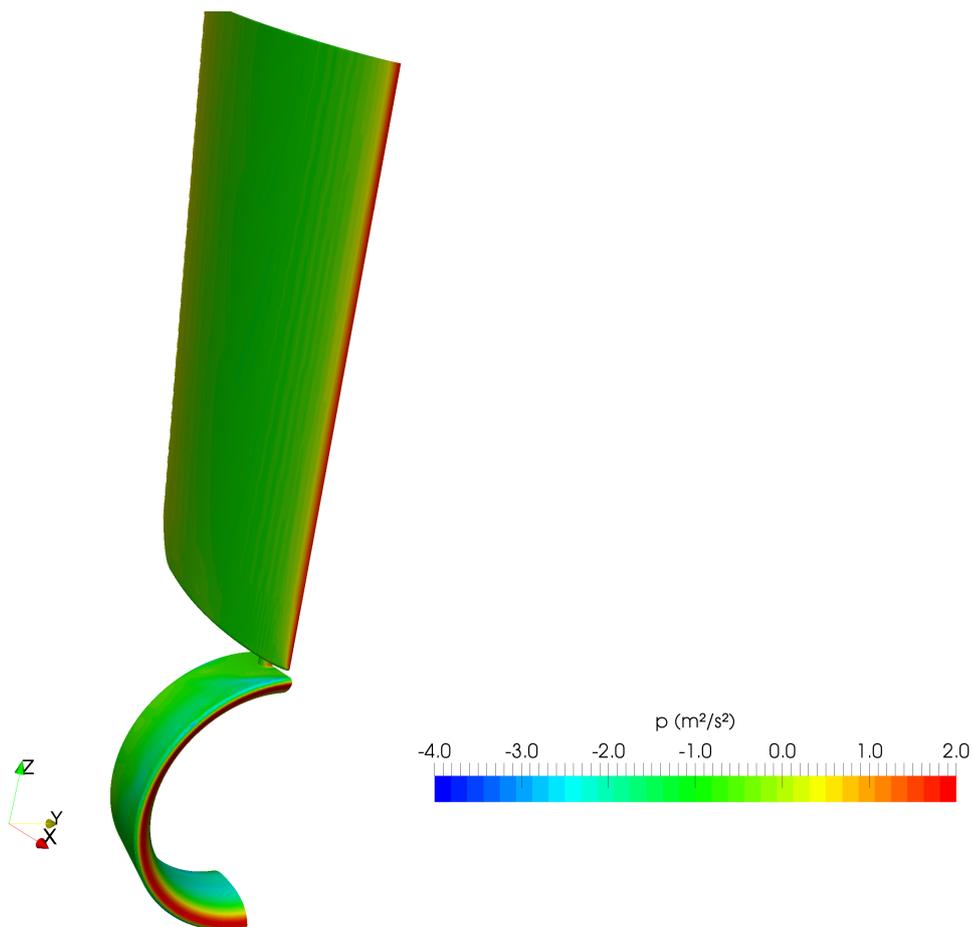


Abbildung 7.16: Druck auf Oberfläche Außenseite - mit Halter

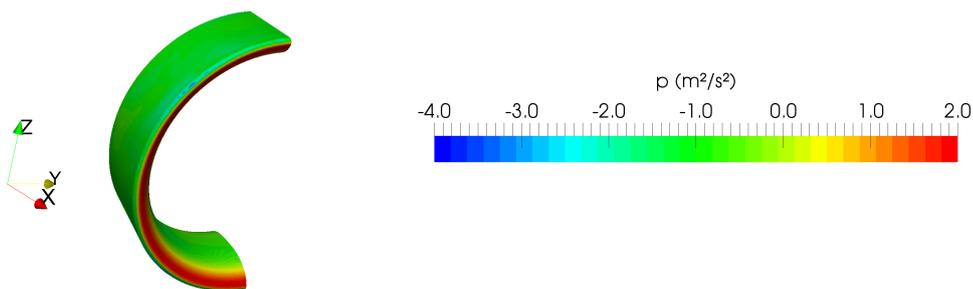


Abbildung 7.17: Druck auf Oberfläche Außenseite - ohne Halter

Die Innenseite der Düse stellt ein Bild dar, dass sich aus den Abbildung 7.14 und Abbildung 7.15 schon schließen ließ. Der Druck fällt an der Innenseite verhältnismäßig stark ab. Aber auch hier stellt man fest, dass die Verteilung nicht über die ganze Fläche homogen verläuft und ebenso stellt man Unterschiede in den Iterationsschritten fest, was Abbildung B.12 bis Abbildung B.15 zeigen.

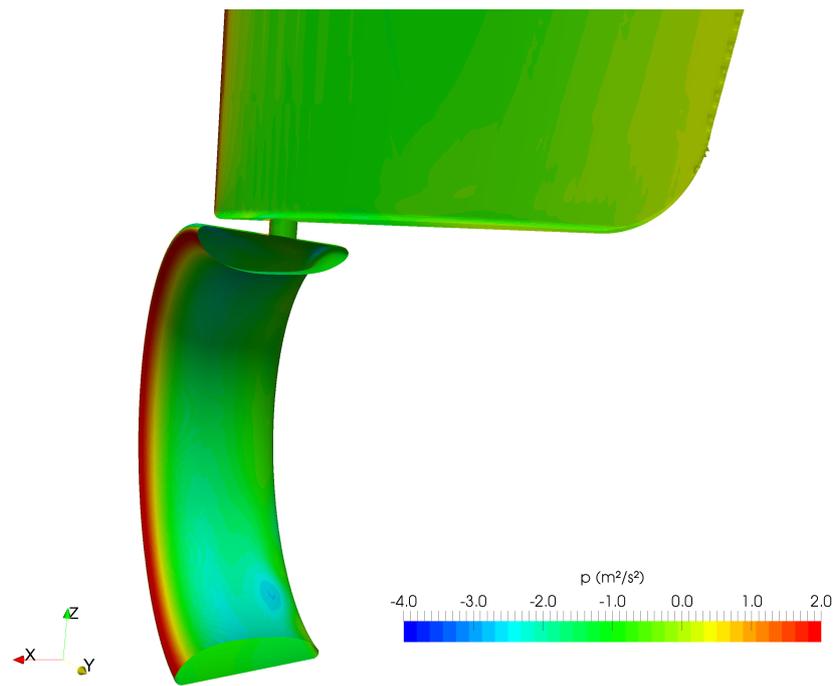


Abbildung 7.18: Druck auf Oberfläche Innenseite - mit Halter

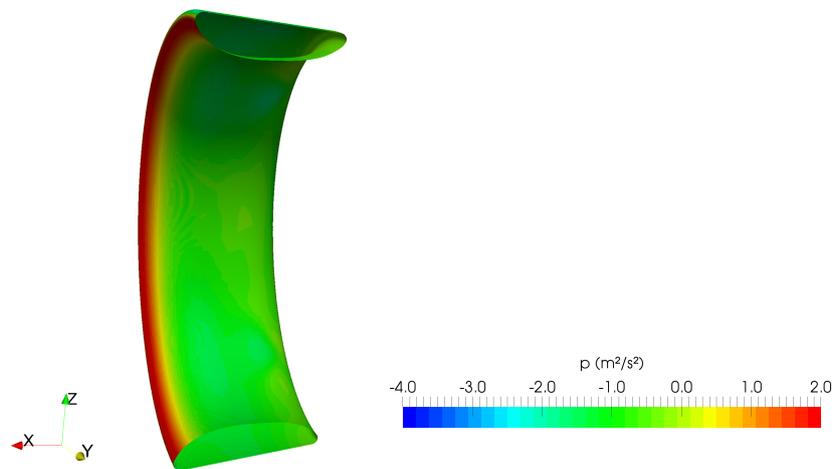


Abbildung 7.19: Druck auf Oberfläche Innenseite - ohne Halter

7.3 Kräfte

Der Vergleich der Kräfte auf die Oberfläche der Düse zeigt keinen stetigen Verlauf. Die Mittelwertbildung ergibt für die Rechnung

	Kraft [N]
mit Haltearm	-13,494
nur Düse	-13,809

Tabelle 7.1: Mittelwerte der Kraft auf die Düse

und in Abbildung 7.20 ist der Verlauf der Kräfte über die Iterationen aufgetragen. Diese stellen sich nicht konstant ein. Vergleicht man die Mittelwerte miteinander, ergibt sich eine Abweichung von 0,01%, was als sehr gering einzuschätzen ist. Dass die Kraft mit Haltearm geringer ist als die mit Düse, entspricht genau dem, was durch die Wirbelbildung zu erwarten ist. Diese verursachen eine Kraft genau entgegengesetzt zur Kraft, die durch die Strömung aufgebracht wird.

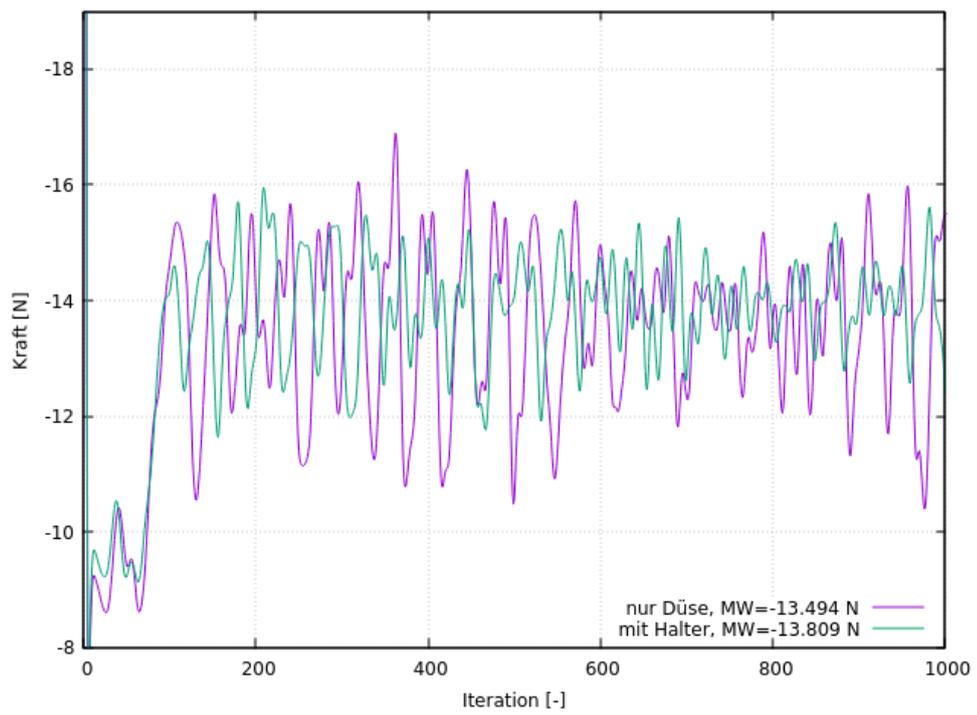
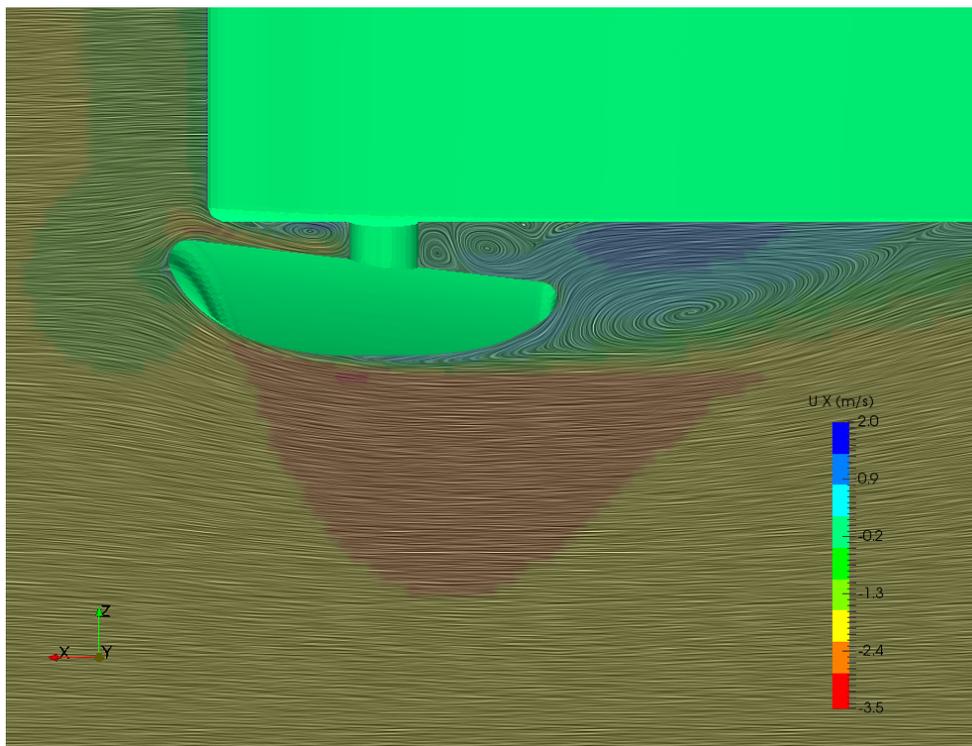
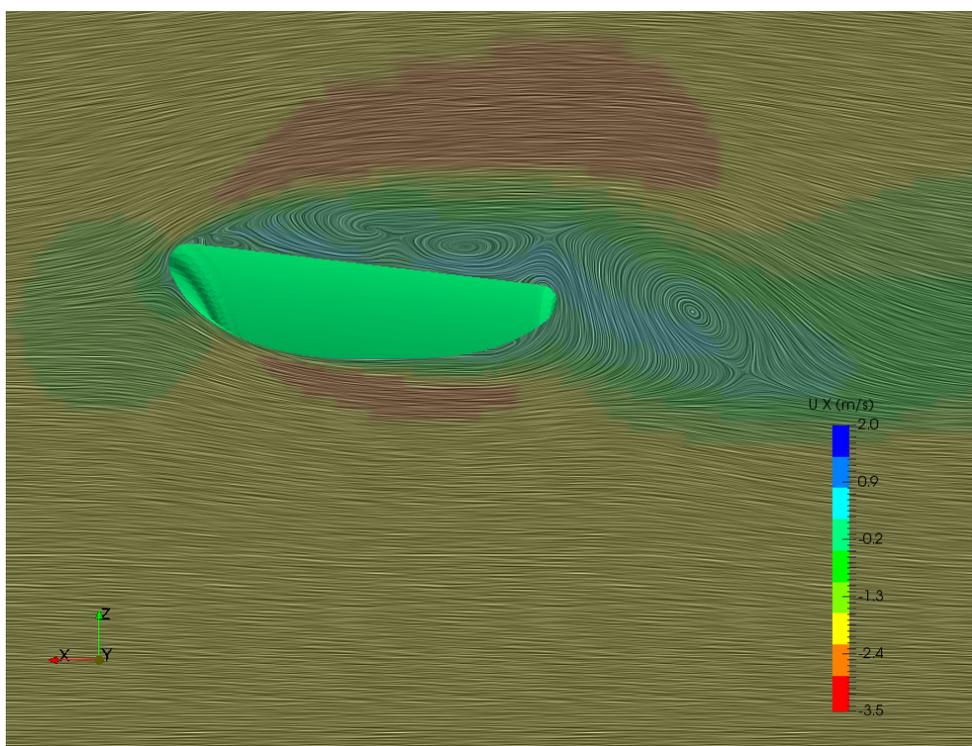


Abbildung 7.20: Kräfte auf die Düse über Iterationen

7.4 Interpolierte Darstellungen

Noch deutlicher werden die Turbulenzen bei einer anderen Darstellung durch *Line Integral Convolution* (LIC, übersetzt Linien Integral Faltung), diese Funktion stellt ein Muster dar, das den Vektorfeldtangente folgt. Hier ist sehr wichtig, dass es sich um eine Interpolation handelt. Das hier verwendete Netz kann solch feine Wirbel, vor allem in Wandnähe, eigentlich nicht auflösen. Diese sind durch die verwendeten Daten interpoliert. Diese Abbildungen zeigen nur einen möglichen Verlauf. Sie werden hier verwendet, weil sie die Wirbel, die man auch schon in den Stromlinien erkennen konnte, etwas bildhafter darstellen. Gerade der Bereich direkt hinter der Verbindung, wo die Kraftmessung stattfindet, weist sehr kleine Wirbel auf. Diese können einen deutlichen Einfluss auf die Kraftmessung haben. Auch der bereits abgebildete große Wirbel kann solche Einflüsse verursachen. Aber auch ohne Haltearm erkennt man hier Verwirbelungen. Diese wären noch genauer zu untersuchen, da sie bei den Stromlinien nicht erkennbar waren. Diese Verwirbelungen können jedoch auch durch den Bernoulli-Ansatz erklärt werden, da sich die Drücke und Geschwindigkeiten ändern.

Abbildung 7.21: Strömung in der xz -Ebene - mit HalterAbbildung 7.22: Strömung in der xz -Ebene - ohne Halter

Schneidet man die Düse in der yz -Ebene, sieht man die unterschiedliche Verteilung der Wirbel um die Düse herum. Jeder Schnitt der Düsen-Normalen würde ein anderes Bild ergeben. Bei der Düse sieht man den Strömungsverlauf zu und von der Düse weg, der im Mittel annähernd symmetrisch anmutet. Bei der Darstellung mit Haltearm sieht man einen Einfluss, der um den Halter-Bereich gewölbt ist. Dieser Schnitt zeigt die Gründe für die auftretende Asymmetrie. Die Wirbel entstehen nicht nur in der Schnittebene der Düse, sondern verlaufen natürlich auch in alle Richtungen. Dadurch herrscht stetig eine Wechselwirkung zwischen den Schnittebenen und es kann keine stationäre Strömung ausgebildet werden. Abbildung 7.23 und Abbildung 7.24 sind mit der Geschwindigkeit in y -Richtung eingefärbt. Diese sollte bestenfalls gleich null sein. Im Bereich der Düse sieht man, dass dies punktuell nicht der Fall ist, jedoch in einem geringen Ausmaß. Jedoch sind diese Geschwindigkeiten quer zur Strömungsrichtung vermutlich ein Verursacher für die transienten Vorgänge. Am Schwert erkennt man auch einen durchgehenden Anteil der Geschwindigkeit in y -Richtung. Dieser war jedoch zu erwarten, da die Form des Schwertes die Strömung entsprechend umlenkt.

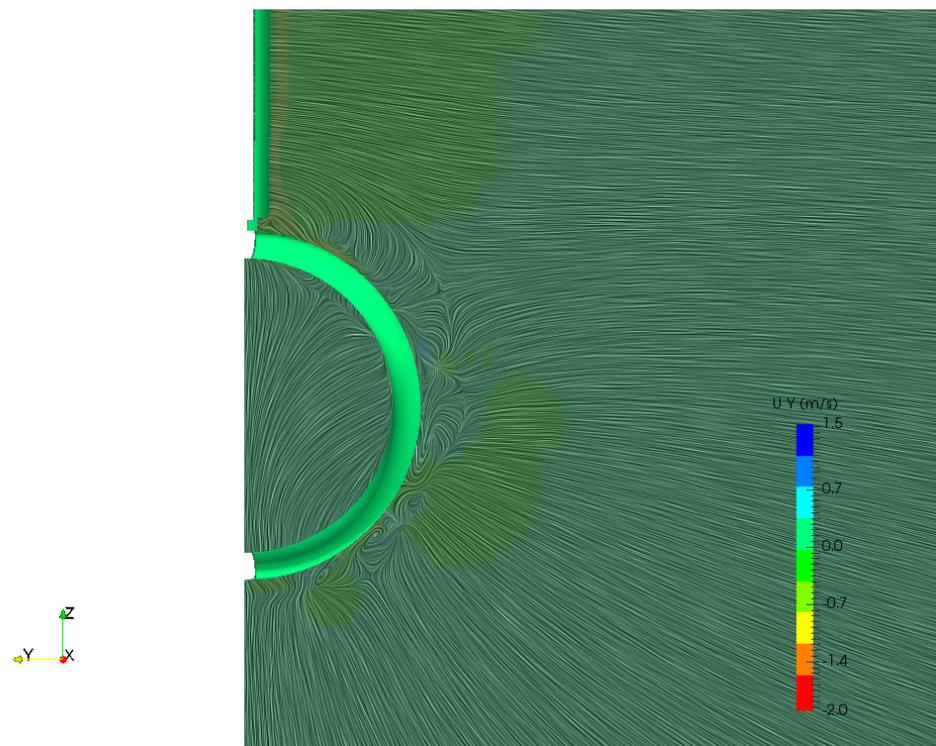


Abbildung 7.23: Strömung um die Düse in der yz-Ebene - mit Halter

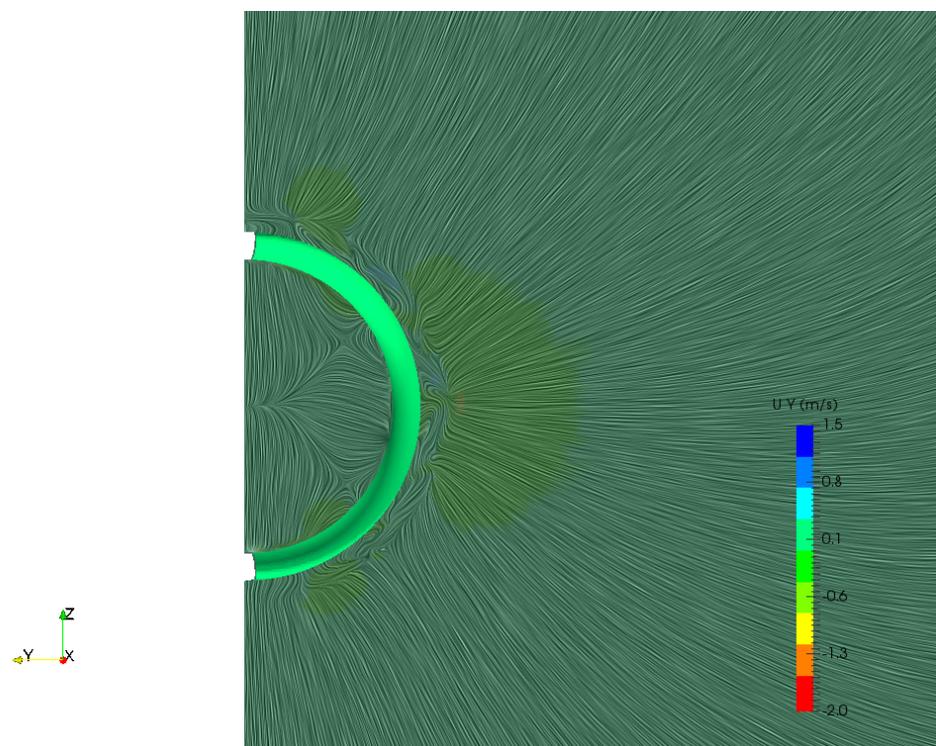


Abbildung 7.24: Strömung um die Düse in der yz-Ebene - ohne Halter

8 Fazit

Urteil über die Qualität des Haltearms

Die Geometrie ist im Prinzip strömungsgünstig geformt. Die Ausführung der Übergänge ist allerdings sehr scharfkantig. Besonders die hintere Ecke des Schwertes fällt hier auf, da zwar versucht wurde, einen fließenden Übergang zur Kante zu realisieren, aber dabei scharfe Kanten zu den Seitenwänden in Kauf genommen wurden. Die hintere Kante ist allgemein scharfkantig. Ob dies auch so gefertigt werden würde ist fraglich, denn durch Fasen oder sogar Schleifen der Hinterkante könnten diese Ecken vermindert werden oder sogar gänzlich wegfallen. Diese Kanten sind bereits für die Vernetzung und Simulation weniger gut geeignet und verursachen erfahrungsgemäß teilweise Strömungsablösungen. Diese wurden in diesem Fall nicht beobachtet, aber die Vernetzung dieser Geometrie war an dieser Stelle problematisch. Dadurch wurde das Gitter in diesem Bereich nicht ideal aufgelöst, was die Lösung in diesem Bereich ungenau werden lässt. Dies wurde bewusst zugelassen, da es für die Entwicklung der Kraft an der Düse als wenig relevant betrachtet wurde.

Besonders auffallend war die ausgeprägte Wirbelbildung direkt am Halter zwischen Düse und Schwert. Hier erfolgt die Kraftmessung und sollte somit möglichst keinen turbulenten Kräften und Verwirbelungen ausgesetzt sein.

In den Berechnungen konnte nur eine sehr geringe Abweichung von 0,01 % festgestellt werden. Allerdings sind diese durch mögliche Fehler im Zusammenhang mit der Art der Simulation nicht ohne Vorbehalt und weitergehende Simulationen zu verwerten. Allerdings entsprechen sie den erwarteten Tendenzen und die Kräfte bei dem Fall mit Haltearm sind geringer. Dies spricht dafür, dass die Wirbelbildung eine Kraft induziert, die gegen die durch die Strömung aufbrachte Kraft wirkt. Durch das Fehlen eines implementierten Propellers kann jedoch nur für den untersuchten Fall eine Aussage getroffen werden. Durch den Propeller könnten die Kräfte und somit auch die Abweichungen höher sein.

Eignung der Simulationsparameter

Wie bereits an verschiedenen Stellen erwähnt wurde, haben die Rechnungen keine als konvergiert anzusehenden Residuen angenommen. Aus diesem Grund wird vermutet, dass es sich bei dem Problem nicht wie eingangs angenommen um ein stationäres, sondern um ein transientes Problem handelt. Die Residuen sind dafür das erste Indiz. Auch die Kraftverläufe nehmen keinen konstanten Wert an, sondern bewegen sich um einen Mittelwert. Ein weiteres Indiz für diese Annahme sind die unterschiedlichen Geschwindigkeitsverläufe an der Düse. Diese in Abbildung 7.23 und Abbildung 7.24 gut sichtbaren

Unregelmäßigkeiten stehen in Wechselwirkung zueinander und können somit keine stationäre Lösung erzielen. Auch die Ausbildung einer Kármánschen Wirbelstraße ist denkbar und könnte in einer stationären Lösung nicht dargestellt werden. Die Simulation wurde nur für den Fall von einer Geschwindigkeit in x-Richtung von 2 m/s durchgeführt. Eine Berechnung mit einer anderen Geschwindigkeit könnte bereits zu einer stationären Lösung führen.

9 Ausblick

Weiterführende Simulationen zu dem neuen Haltearm sind eindeutig angezeigt und sollten unbedingt transient erfolgen. Eine Simulation mit einem vereinfachten Propellermodell wäre anzuraten, da die auftretenden Kräfte größer werden und im Zweifel eine größere prozentuale Abweichung verursachen könnten. Die Implementierung einer *fixedJump* Randbedingung scheint vielversprechend zu sein und sollte weiter verfolgt werden.

Eine Überarbeitung der konstruktiven Ausführung könnte zu einer vergleichenden Simulation herangezogen werden und somit die bestmögliche Geometrie gefunden werden. Hier wäre eine Möglichkeit, den Halter zu verlängern. Dies würde aber vermutlich wegen des vergrößerten Hebelarms zu viel größeren Kräften auf der üblicherweise genutzten Kraftmessdose führen und möglicherweise die maximal zulässige Belastung überschreiten. Eine andere Möglichkeit wäre eine konstruktive Anpassung des Übergangs von dem Haltearm zur Düse. Dies würde jedoch eventuell zu Spalten führen oder Manipulation an den Düsen nötig machen, was wegen der präzisen Handarbeit der Einzelstücke wenig sinnvoll erscheint.

Die in dieser Arbeit simulierten Werte deuten auf einen recht geringen Einfluss des Haltearms auf die Kraftmessung hin, jedoch muss dies mindestens mit einer transienten Rechnung nachgewiesen werden. Eine weitere zu empfehlende Maßnahme nach der Inbetriebnahme des Haltearms am Freifahrtgerät H39 ist eine Validierung mittels echter Freifahrtversuche im Schlepptank. Hierzu können die Daten aus den CFD-Rechnungen, als auch aus Messungen mit dem Freifahrtgerät H32 herangezogen werden.

Auch die Parameter der Berechnung sollten erweitert werden. Hier wurde die Rechnung nur mit einer Geschwindigkeit von 2 m/s durchgeführt. Der Schleppwagen kann Geschwindigkeiten bis zu 10 m/s liefern. Weitere Geschwindigkeiten könnten bereits andere Ergebnisse bezüglich einer transienten oder stationären Lösung liefern. Allerdings bleibt die Empfehlung, die Rechnung transient durchzuführen, da dies die möglichen instationären Vorgänge immer auflöst.

10 Zusammenfassung

In der vorliegenden Arbeit wurden mittels des CFD-Programms OpenFOAM numerische Strömungssimulationen durchgeführt. Die Problemstellung war, dass ein Freifahrtgerät mit einem Haltearm ausgestattet werden soll, der eine Düse aufnehmen kann. Ziel dessen ist es, die Kraft auf die Düse bestimmen zu können, wenn ein Propeller in einer Düse betrieben wird. Hierfür wurde mit dem Vernetzungsprogramm *snappyHexMesh* ein numerisches Gitter erzeugt, das die Geometrie in ein mit *blockMesh* generiertes Hintergrundgitter einfügt. Die Randbedingungen und Simulationsparameter wurden gemäß OpenFOAM Struktur in Dateien festgelegt. Das Post-Processing wurde mit *paraFoam* durchgeführt. Im Verlauf der Berechnungen wurde deutlich, dass die in der Aufgabenstellung vorgegebenen Annahmen einer stationären Berechnung möglicherweise nicht zielführend sind. Die vorliegenden Ergebnisse wurden mit Bedacht auf die möglicherweise inkorrekten Randbedingungen interpretiert. Die Ergebnisse zeigen eine Tendenz, dass der Haltearm aufgrund von Rückströmungen in Form von Wirbeln einen geringen Einfluss auf die Messergebnisse hat. Die Ergebnisse sind zu überprüfen, sie zeigen aber ein Strömungsverhalten, das zu erwarten war. Es wurde empfohlen, die Rechnungen mit einem vereinfachten Propellermodell zu simulieren, da es hier zu erhöhten Kräften kommen wird und sich dadurch die Größenordnung der Abweichung ändern kann. Es wurde gezeigt, dass die Funktion einer *actuatorDisk* für die Simulation von Schiffspropellern ungeeignet ist und eine Alternative in Form der *fixedJump*-Randbedingung wurde aufgezeigt. Außerdem sollte die Simulation mit einem transienten Solver geprüft werden. Eine Validierung der CFD-Rechnung mit Versuchsergebnissen der HSVA ist ebenso zu empfehlen. Außerdem ist es anzuraten, die Geometrie auf verbesserte Strömungseigenschaften zu prüfen und ggf. verschiedene Ansätze zu vergleichen, wobei die Grundform des Haltearms bereits eine strömungsgünstige Form aufweist.

Literatur

- [1] Böhle, M., Oertel Jr., H. und Reviol, T.: Strömungsmechanik. Springer Vieweg, 2015.
- [2] Clauss, G.: Schiffs- und meerestechnisches Versuchswesen. Technische Universität Berlin.
- [3] Eggleston, D. M. und Stoddard, F. S.: Wind turbine engineering design. Van Nostrand Reinhold Company, 1987.
- [4] Ferziger, J. H. und Perić, M.: Numerische Strömungsmechanik. Springer-Verlag, 2008.
- [5] Greenshields, C. J.: OpenFoam User Guide. 2017.
- [6] Gschaider, B.: blockMesh grading calculation. Adresse: https://openfoamwiki.net/index.php/Scripts/blockMesh_grading_calculation (besucht am 10.05.2018).
- [7] Hamburgische Schiffbau-Versuchsanstalt GmbH: Interne Versuchsunterlagen.
- [8] Hamburgische Schiffbau-Versuchsanstalt GmbH: Firmen Homepage. Adresse: <https://www.hsva.de/> (besucht am 18.04.2018).
- [9] International Towing Tank Conference: Recommended Procedures and Guidelines - Practical Guidelines for Ship CFD Applications. 2014.
- [10] International Towing Tank Conference: Uncertainty Analysis in CFD - Uncertainty Assessment Methodology. 1999.
- [11] Kornev, N.: Propellertheorie. 2009. Adresse: https://www.lemos.uni-rostock.de/fileadmin/uni-rostock/Alle_MSF/Lemos/Lehre/Sommersemester/Propeller/Prop_new.pdf (besucht am 18.04.2018).
- [12] Krueger, S.: Schiffspropeller. 2005. Adresse: <https://www.tuhh.de/t3resources/ssi/layout01INSTITUTE/Lehre/Schiffspropeller/Vorlesungsunterlagen/Propellertheorie.pdf> (besucht am 18.04.2018).
- [13] Laurien, E. und Oertel Jr., H.: Numerische Strömungsmechanik. Springer Vieweg, 2013.
- [14] Menter, F. R.: Zonal Two Equation $k-\omega$ Turbulence Models for Aerodynamic Flows. Eloret Institute, Sunnyvale, 1993.
- [15] Rotteveel, E.: A ducted actuator disk. Adresse: <https://www.erikrotteveel.com/openfoam/propeller-modeling/a-ducted-actuator-disk/> (besucht am 17.05.2018).

- [16] Wulf, P.: Vorlesungsunterlagen der Kurse CFD-BS und CFD-EA. Hochschule für angewandte Wissenschaften Hamburg, 2015.

A Software-Einstellungen

In diesem Teil des Anhangs sollen die OpenFoam Dateien aufgeführt werden. Diese werden gemäß der Case-Struktur in OpenFOAM sortiert. Die Dateien sind aus der Berechnung mit Haltearm mit 2.1M Zellen entnommen.

A.1 system

A.1.1 blockMeshDict

```

/*-----*- C++ -*-----*\
| ===== |
| \\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      /  O peration  | Version: 4.1 |
|  \\    /   A nd        | Web:      www.OpenFOAM.org |
|   \\  /    M anipulation | |
\*-----*-*/
FoamFile
{
version      2.0;
format       ascii;
class        dictionary;
object       blockMeshDict;
}

// * * * * *

convertToMeters 0.001;

vertices
(
//Block Inlet
(2600 0 -680) //0
(200 0 -680) //1
(200 -750 -680) //2
(2600 -750 -680) //3
(2600 0 709) //4

```

```
(200 0 709) //5
(200 -750 709) //6
(2600 -750 709) //7

//Block Mtte
(200 0 -680) //8
(-500 0 -680) //9
(-500 -750 -680) //10
(200 -750 -680) //11
(200 0 709) //12
(-500 0 709) //13
(-500 -750 709) //14
(200 -750 709) //15

//Block Outlet
(-500 0 -680) //16
(-3700 0 -680) //17
(-3700 -750 -680) //18
(-500 -750 -680) //19
(-500 0 709) //20
(-3700 0 709) //21
(-3700 -750 709) //22
(-500 -750 709) //23
);

blocks
(
hex (0 1 2 3 4 5 6 7) (48 25 70) simpleGrading (0.2 2.14 1)
hex (1 9 10 2 5 13 14 6) (35 25 70) simpleGrading (1 2.14 1)
hex (9 17 18 10 13 21 22 14) (64 25 70) simpleGrading (5.02 2.14 1)

);

edges
(
);

boundary
(

inlet
{
type patch;
```

```
faces
(
(0 4 7 3)
);
}
outlet
{
type patch;
faces
(
(17 18 22 21)
);
}

wand
{
type symmetry;
faces
(
(3 7 6 2)
(4 5 6 7)
(0 3 2 1)

(1 9 10 2)
(2 6 14 10)
(5 13 14 6)

(9 17 18 10)
(10 18 22 14)
(13 21 22 14)
);
}

symmetrie
{
type symmetry;
faces
(
(1 5 4 0)
(9 13 5 1)
(17 21 13 9)
);
}
);
```

```
// ***** //
```

A.1.2 controlDict

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 4.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n | |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  class dictionary;
  object controlDict;
}
// ***** //

application simpleFoam;

startFrom latestTime;

startTime 0;

stopAt endTime;

endTime 1000;

deltaT 1;

writeControl timeStep;

writeInterval 100;

purgeWrite 0;

writeFormat binary;

writePrecision 6;
```

```
writeCompression uncompressed;
```

```
timeFormat      general;
```

```
timePrecision   6;
```

```
runTimeModifiable true;
```

```
functions
{
```

```
#include "forces1"
#includeFunc yPlus
}
```

```
// ***** //
```

A.1.3 decomposeParDict

```
/*-----*- C++ -*-----*\
| ===== |
| \\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      /  O peration  | Version: 4.1 |
|  \\    /   A nd        | Web:      www.OpenFOAM.org |
|   \\  /    M anipulation |
\*-----*/
```

```
FoamFile
{
version      2.0;
format       ascii;
class        dictionary;
object       decomposeParDict;
}
```

```
// * * * * * //
```

```
numberOfSubdomains 4;
```

```
method        simple;
// method     ptscotch;
```

```
simpleCoeffs
{
```

```
n          (2 2 1);
delta      0.001;
}
```

```
hierarchicalCoeffs
{
n          (1 1 1);
delta      0.001;
order      xyz;
}
```

```
manualCoeffs
{
dataFile   "";
}
```

```
// ***** //
```

A.1.4 forces1

```
/*-----* C++ *-----*\
| ===== |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 4.1 |
|  \\    / A nd         | Web:      www.OpenFOAM.org |
|   \\// M anipulation  | |
\*-----*/
```

```
forces1
{
type      forces;

libs ( "libforces.so" );

writeControl  timeStep;
timeInterval  1;

log         yes;

patches     ( "duese" );
rho         rhoInf;      // Indicates incompressible
log         true;
rhoInf      1000;        // Redundant for incompressible
```

```

porosity      no;

CofR          (0 0 0);    // Rotation around centre line of propeller
pitchAxis     (0 1 0);
}

```

```
// ***** //
```

A.1.5 fvSchemes

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O p e r a t i o n | Version: 4.1 |
|  \\    / A n d           | Web:      www.OpenFOAM.org |
|   \\  / M a n i p u l a t i o n |
\*-----*-

```

```
FoamFile
```

```

{
version      2.0;
format       ascii;
class        dictionary;
object       fvSchemes;
}

```

```
// * * * * * *
```

```
ddtSchemes
```

```

{
default      steadyState;
}

```

```
gradSchemes
```

```

{
default      Gauss linear;
//   grad(U)      cellLimited Gauss linear 1;
}

```

```
divSchemes
```

```

{
default      none;
div(phi,U)   bounded Gauss linearUpwind grad(U);
div(phi,k)   bounded Gauss upwind;
div(phi,omega) bounded Gauss upwind;
}

```

```
div((nuEff*dev2(T(grad(U)))) Gauss linear;
}
```

```
laplacianSchemes
{
default          Gauss linear corrected;
}
```

```
interpolationSchemes
{
default          linear;
}
```

```
snGradSchemes
{
default          corrected;
}
```

```
wallDist
{
method meshWave;
}
```

```
// ***** //
```

A.1.6 fvSolution

```
/*-----*- C++ -*-----*\
| ===== |
| \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O p e r a t i o n | Version: 4.1 |
|  \\    /   A n d          | Web:      www.OpenFOAM.org |
|   \\  /    M a n i p u l a t i o n |
\*-----*\
FoamFile
{
version      2.0;
format       ascii;
class        dictionary;
object       fvSolution;
}
// * * * * * *
```

```
solvers
{
p
{
solver          GAMG;
tolerance       1e-7;
relTol          0.01;
smoother        GaussSeidel;
}

Phi
{
$P;
}

U
{
solver          smoothSolver;
smoother        GaussSeidel;
tolerance       1e-8;
relTol          0.1;
nSweeps         1;
}

k
{
solver          smoothSolver;
smoother        GaussSeidel;
tolerance       1e-8;
relTol          0.1;
nSweeps         1;
}

omega
{
solver          smoothSolver;
smoother        GaussSeidel;
tolerance       1e-8;
relTol          0.1;
nSweeps         1;
}
}

SIMPLE
```

```
{
nNonOrthogonalCorrectors 0;
consistent yes;
}
```

```
relaxationFactors
```

```
{
equations
{
U          0.9;
k          0.7;
omega      0.7;
}
}
```

```
cache
```

```
{
grad(U);
}
```

```
// ***** //
```

A.1.7 meshQualityDict

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 4.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
```

```
FoamFile
```

```
{
version      2.0;
format       ascii;
class        dictionary;
location     "system";
object       meshQualityDict;
}
```

```
maxNonOrtho 80;
maxBoundarySkewness 20;
maxInternalSkewness 4;
```

```

maxConcave 180;
minVol -1e30;
minTetQuality 1e-13;
minArea -1;
minTwist -1E15;
minDeterminant 0.0005;
minFaceWeight 0.001;
minVolRatio 0.001;
minTriangleTwist -1;
nSmoothScale 10;
errorReduction 0.85;

relaxed
{
maxNonOrtho 120;
}

```

A.1.8 snappyHexMeshDict

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 4.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
FoamFile
{
version 2.0;
format ascii;
class dictionary;
object snappyHexMeshDict;
}
// * * * * * //

castellatedMesh true;
snap true;
addLayers false;

```

```
//hier die Einstellungen für Verfeinerung und Layer
nD 4; //Verfeinerung Düse 4
nH 3; //Verfeinerung 3
l 3; //Layer
nR 4; //Verfeinerung Halter und Engstelle
nZ 3; //Verfeinerung Zylinder (um Düse herum und stromab)
```

```
geometry
{
halter_fein_1_blender.stl //oberteil (mit layern)
{
type triSurfaceMesh;
name schwert;
}
halter_fein_2_blender.stl //kraftmessdose (ohne layer)
{
type triSurfaceMesh;
name halter;
}
```

```
duese.stl // (mit layern)
{
type triSurfaceMesh;
name duese;
}
```

```
refinementBox
{
type searchableBox;
min ( -0.30 -0.04 0.13);
max (0.04 0 0.20);
}
```

```
refinementSphere
{
type searchableSphere;
centre (-0.005 0 0.133);
radius 0.015;
}
```

```
refinementCylinder
{
type searchableCylinder;
```

```
point1      (0.08 0 0);
point2      (-0.4 0 0);
radius      0.17;
}

};

castellatedMeshControls
{
maxLocalCells 100000;
maxGlobalCells 10000000;
minRefinementCells 10;
maxLoadUnbalance 0.10;
nCellsBetweenLevels 3;

features (
{
file "feature_edges_half.eMesh";
level $nH;
}
);

refinementSurfaces
{
duese { level ($nD $nD); }
schwert { level ($nH $nH); }
halter { level ($nD $nD); }

}

resolveFeatureAngle 15;

refinementRegions
{
refinementBox
{
mode inside;
levels (($nR $nR));
}

refinementCylinder
{
mode inside;
```

```
levels (($nZ $nZ));  
}
```

```
refinementSphere  
{  
mode inside;  
levels (($nD $nD));  
}
```

```
}  
locationInMesh (1 0 0);  
allowFreeStandingZoneFaces true;  
}
```

```
snapControls  
{  
nSmoothPatch 5;  
tolerance 2.0;  
nSolveIter 30;  
nRelaxIter 5;  
nFeatureSnapIter 10;  
implicitFeatureSnap false;  
explicitFeatureSnap true;  
multiRegionFeatureSnap true;  
}
```

```
addLayersControls  
{  
relativeSizes false;  
layers  
{  
"duese" { nSurfaceLayers 5; }  
"schwert" { nSurfaceLayers 5; }  
  
}
```

```
expansionRatio 1.2;  
firstLayerThickness 0.0001;  
minThickness 0.0001;  
nGrow 1;
```

```

featureAngle 30;
slipFeatureAngle 15;
nRelaxIter 10;
nSmoothSurfaceNormals 100;
nSmoothNormals 100;
nSmoothThickness 150;
maxFaceThicknessRatio 0.8;
maxThicknessToMedialRatio 0.9;
minMedianAxisAngle 130;
nBufferCellsNoExtrude 0;
nLayerIter 15;
}

```

```

meshQualityControls
{
#include "meshQualityDict"
}

```

```

writeFlags
(
scalarLevels
layerSets
layerFields
);

```

```
mergeTolerance 1e-6;
```

```
// ***** //
```

A.2 constant

A.2.1 transportProperties

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 4.1 |
|  \\    / A nd         | Web:      www.OpenFOAM.org |
|   \\  / M anipulation | |
\*-----*-*/
FoamFile

```


A.3 0.Orig

A.3.1 initialConditions

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 4.1 |
|  \\    / A nd         | Web:      www.OpenFOAM.org |
|   \\  / M anipulation |
|-----*\

```

```

flowVelocity      (-2 0 0);
pressure          0;
turbulentKE       0.06;
turbulentOmega    24.5;

```

```
// ***** //
```

A.3.2 fixedInlet

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 4.1 |
|  \\    / A nd         | Web:      www.OpenFOAM.org |
|   \\  / M anipulation |
|-----*\

```

```

inlet
{
type fixedValue;
value $internalField;
}

```

```
// ***** //
```

A.3.3 U

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 4.1 |
|  \\    / A nd         | Web:      www.OpenFOAM.org |
|   \\  / M anipulation |
|-----*\

```

```
\*-----*/
FoamFile
{
version      2.0;
format       ascii;
class        volVectorField;
location     "0";
object       U;
}
// * * * * * //

#include      "include/initialConditions"

dimensions   [0 1 -1 0 0 0 0];

internalField  uniform $flowVelocity;

boundaryField
{
//- Set patchGroups for constraint patches
#includeEtc "caseDicts/setConstraintTypes"

#include "include/fixedInlet"

outlet
{
type          inletOutlet;
inletValue    uniform (0 0 0);
value         $internalField;
}

wand
{
type          symmetry;
}

symmetrie
{
type          symmetry;
}

halter
{
type noSlip;
}
```

```

}

schwert
{
type noSlip;
}

duese
{
type noSlip;
}
}

```

```
// ***** //
```

A.3.4 p

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 4.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
FoamFile
{
version 2.0;
format ascii;
class volScalarField;
object p;
}
// ***** //

#include "include/initialConditions"

dimensions [0 2 -2 0 0 0 0];

internalField uniform $pressure;

boundaryField
{

//- Set patchGroups for constraint patches

```

```
#includeEtc "caseDicts/setConstraintTypes"
```

```
inlet
```

```
{
```

```
type          zeroGradient;
```

```
}
```

```
outlet
```

```
{
```

```
type          fixedValue;
```

```
value         $internalField;
```

```
}
```

```
wand
```

```
{
```

```
type          symmetry;
```

```
}
```

```
symmetrie
```

```
{
```

```
type          symmetry;
```

```
}
```

```
halter
```

```
{
```

```
type zeroGradient;
```

```
}
```

```
schwert
```

```
{
```

```
type zeroGradient;
```

```
}
```

```
duese
```

```
{
```

```
type zeroGradient;
```

```
}
```

```
}
```

```
// ***** //
```



```
wand
{
type      symmetry;
}

halter
{
type      omegaWallFunction;
value     $internalField;
}

schwert
{
type      omegaWallFunction;
value     $internalField;
}

duese
{
type      omegaWallFunction;
value     $internalField;
}

}
```

```
// ***** //
```

A.3.6 nut

```
/*-----*- C++ -*-----*\
| ===== |
|  \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
|  \ \      /  O p e r a t i o n | Version: 4.1 |
|   \ \ /    A n d      | Web:      www.OpenFOAM.org |
|    \ \ /    M a n i p u l a t i o n |
\*-----*-*/
FoamFile
{
version      2.0;
format       ascii;
class        volScalarField;
```



```
{
type      nutkWallFunction;
value     uniform 0;
}
```

```
duese
{
type      nutkWallFunction;
value     uniform 0;
}
}
```

```
// ***** //
```

A.3.7 k

```
/*----- C++ -----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 4.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
```

```
FoamFile
```

```
{
version      2.0;
format       ascii;
class        volScalarField;
object       k;
}
```

```
// ***** //
```

```
#include "include/initialConditions"
```

```
dimensions [0 2 -2 0 0 0 0];
```

```
internalField uniform $turbulentKE;
```

```
boundaryField
```

```
{
//- Set patchGroups for constraint patches
#includeEtc "caseDicts/setConstraintTypes"
```

```
inlet
{
type    fixedValue;
value   0.00135;
}

outlet
{
type            inletOutlet;
inletValue      $internalField;
value           $internalField;
}
symmetrie
{
type            symmetry;
}

wand
{
type            symmetry;
}

halter
{
type            kqRWallFunction;
value           $internalField;
}

schwert
{
type            kqRWallFunction;
value           $internalField;
}

duese
{
type            kqRWallFunction;
value           $internalField;
}

}

// ***** //
```

A.4 Skripte

A.4.1 mesh

```
#!/bin/sh
cd ${0%/*} || exit 1    # Run from this directory

# Löschen vom polyMesh Ordner
# Source tutorial clean functions
. $WM_PROJECT_DIR/bin/tools/CleanFunctions

rm -r constant/polyMesh
rm log.*
rm -rf processor*

#blockmesh und snappy ausführen
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

runApplication blockMesh

runApplication decomposePar
runParallel snappyHexMesh -overwrite

runApplication reconstructParMesh -constant

#-----
```

A.4.2 rechnen

```
#!/bin/sh
cd ${0%/*} || exit 1    # Run from this directory

. $WM_PROJECT_DIR/bin/tools/CleanFunctions

rm log.*
rm -rf processor*
rm -rf postProcessing
foamListTimes -rm

# Source tutorial run functions
```

```
. $WM_PROJECT_DIR/bin/tools/RunFunctions
```

```
runApplication decomposePar
```

```
##- For parallel running
```

```
ls -d processor* | xargs -I {} rm -rf ./{/}/0
```

```
ls -d processor* | xargs -I {} cp -r 0.orig ./{/}/0
```

```
runParallel $(getApplication)
```

```
runApplication reconstructParMesh -constant
```

```
runApplication reconstructPar
```

```
##-----
```


B.2 Abbildungen zu Kapitel 7

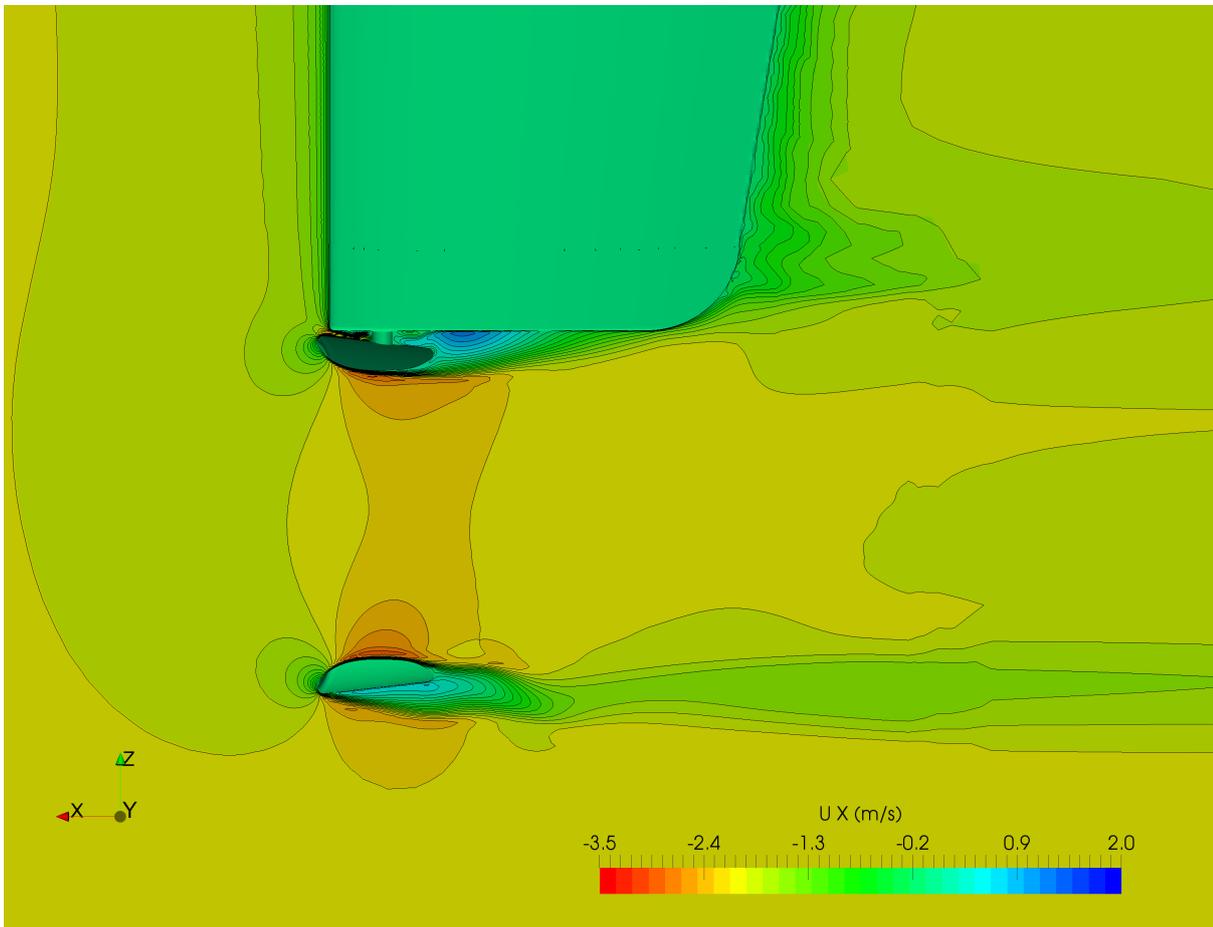


Abbildung B.2: Geschwindigkeit in x-Richtung - Detailansicht - mit Halter - Iteration 700

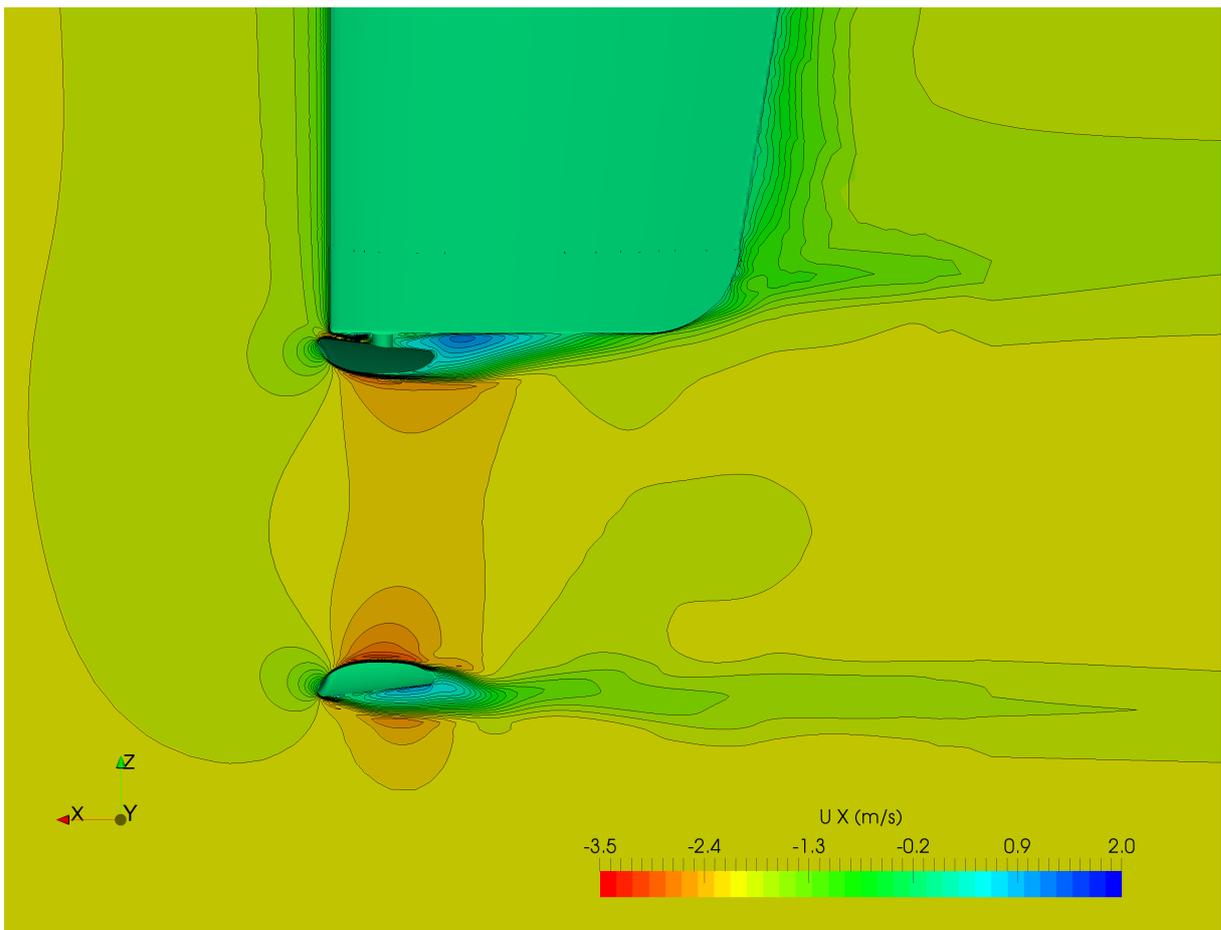


Abbildung B.3: Geschwindigkeit in x-Richtung - Detailansicht - mit Halter - Iteration 800

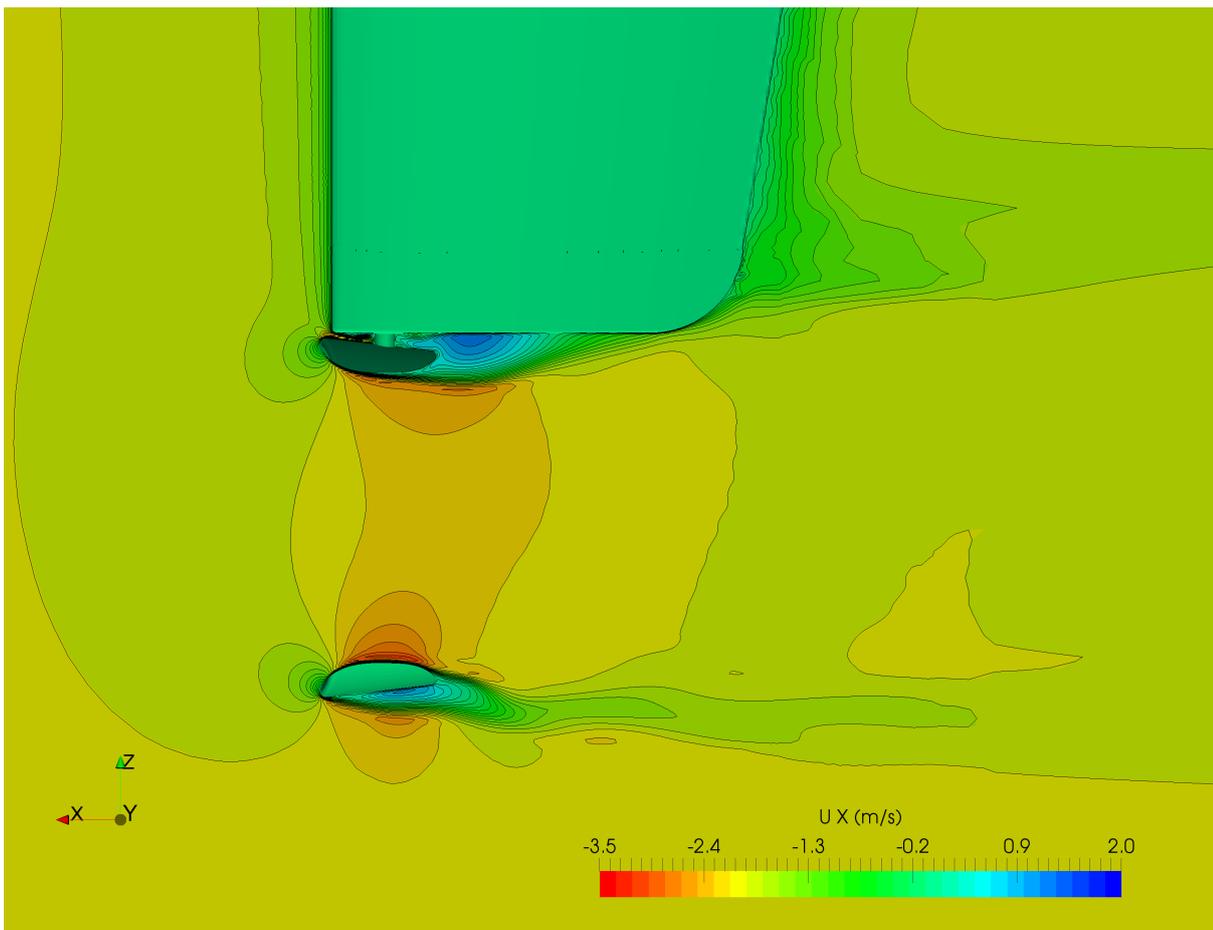


Abbildung B.4: Geschwindigkeit in x-Richtung - Detailansicht - mit Halter - Iteration 900

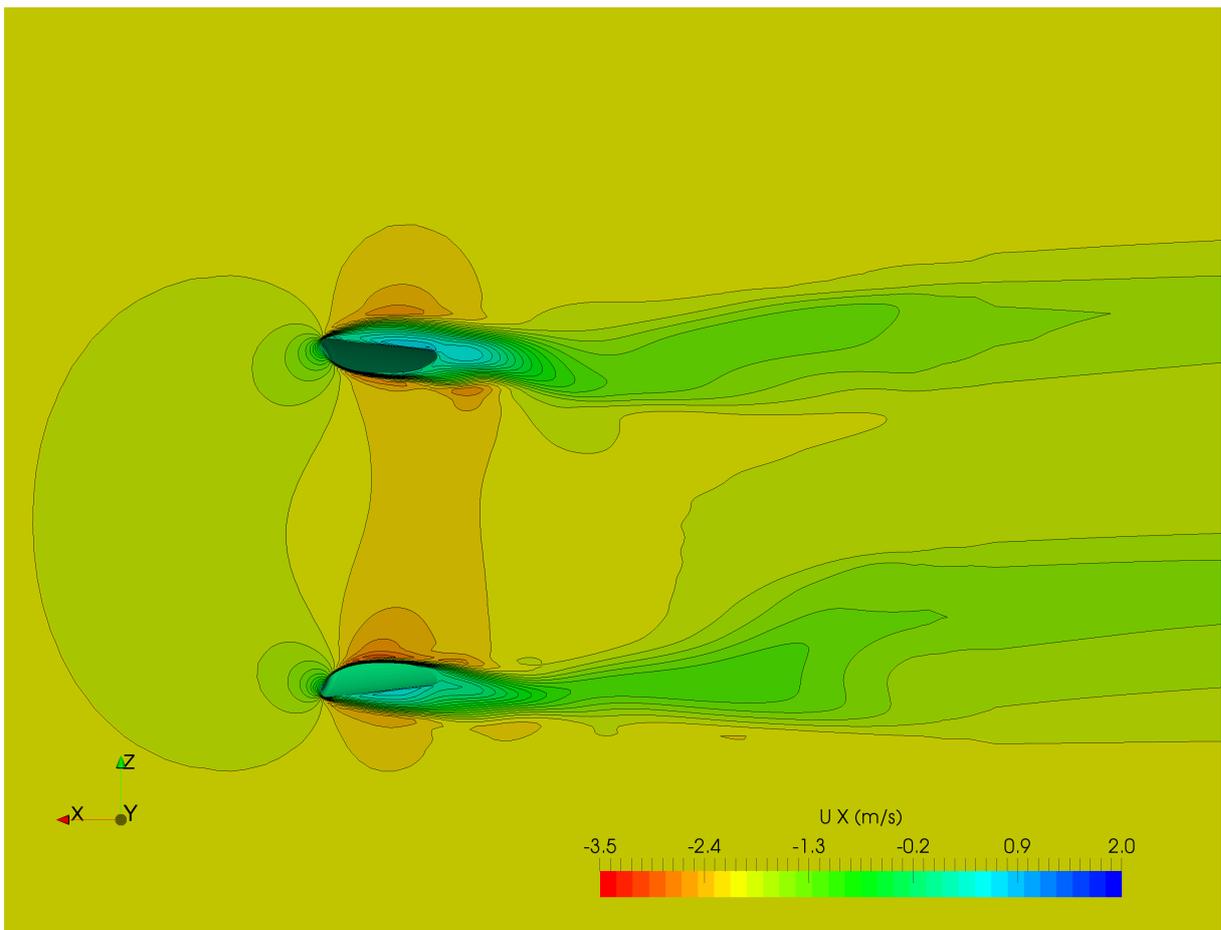


Abbildung B.5: Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter - Iteration 700

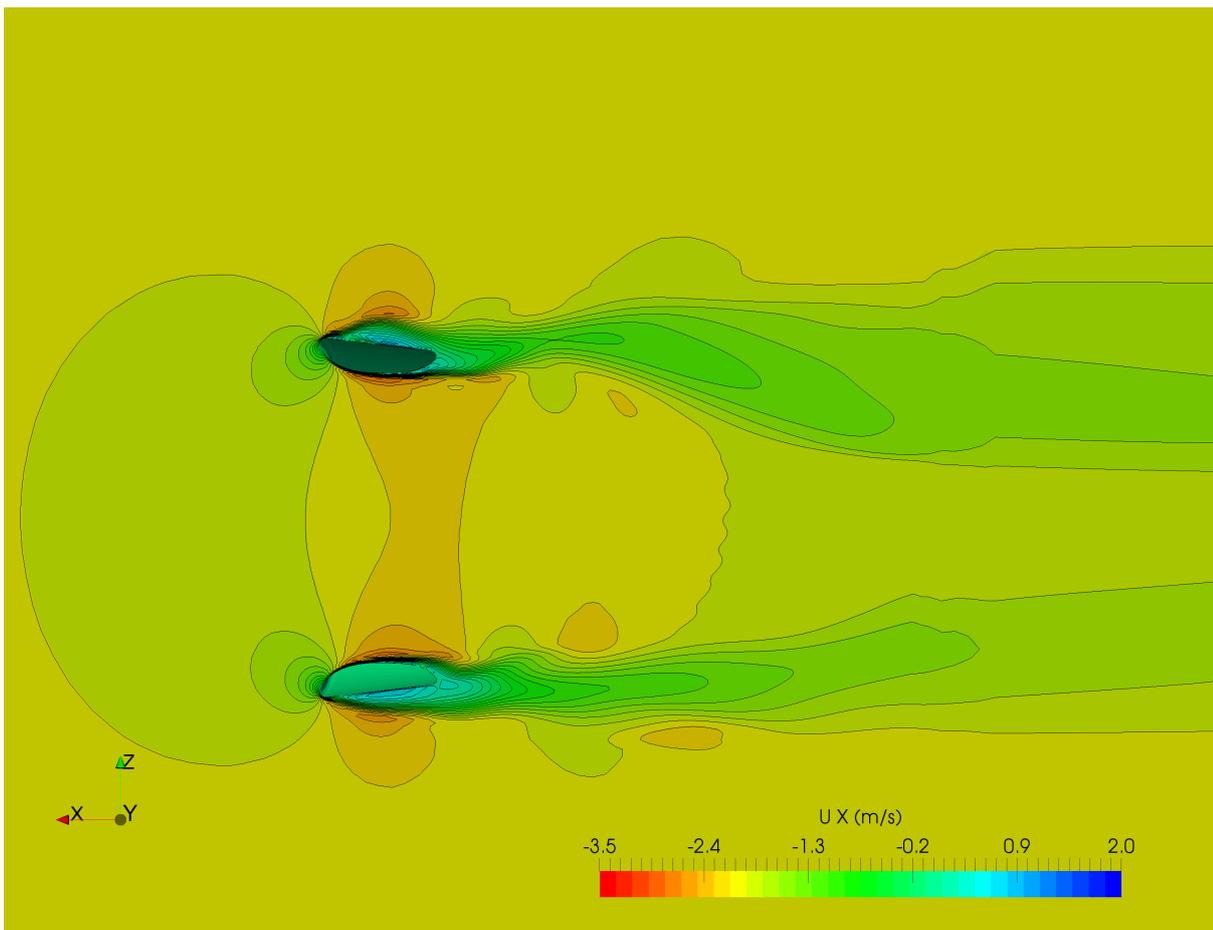


Abbildung B.6: Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter - Iteration 800

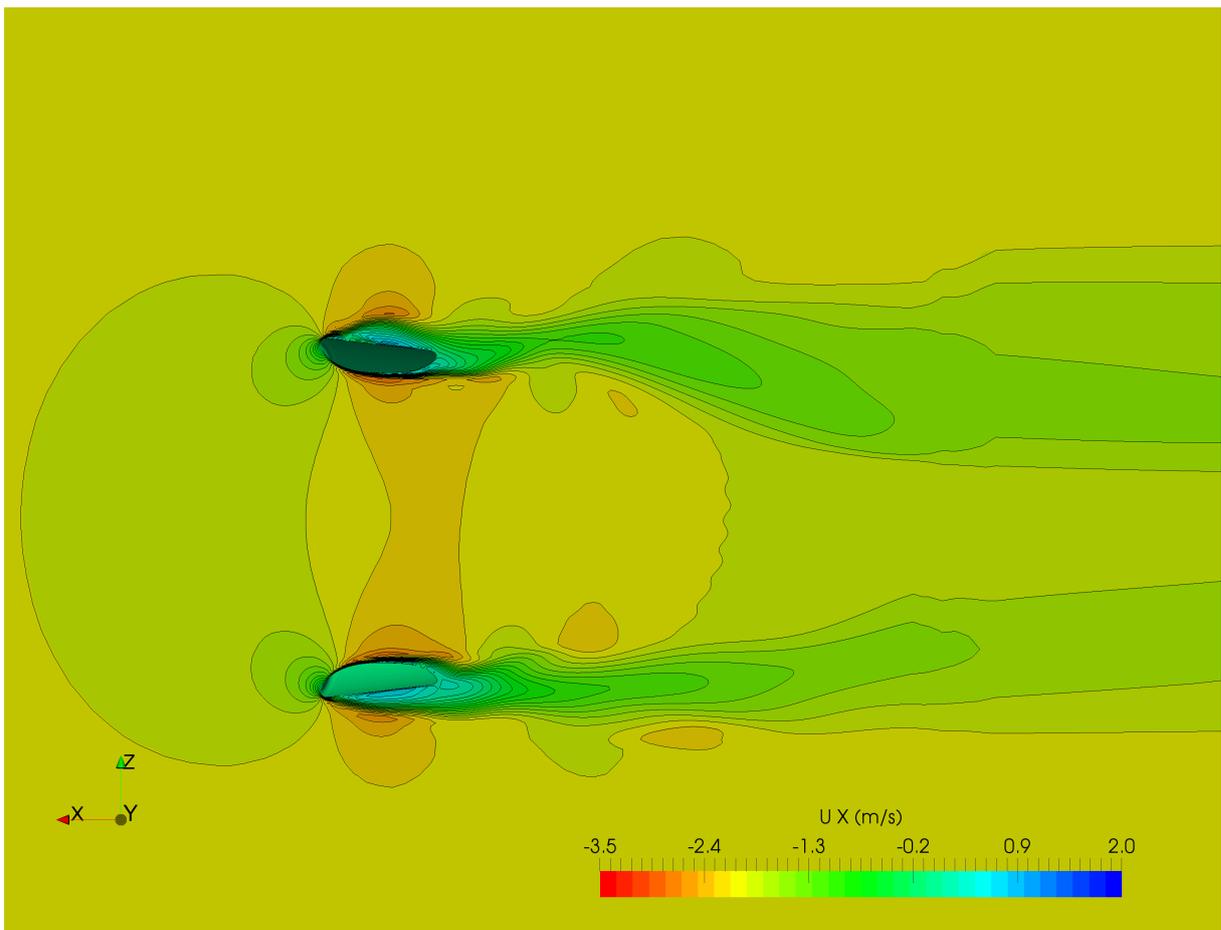


Abbildung B.7: Geschwindigkeit in x-Richtung - Detailansicht - ohne Halter - Iteration 900

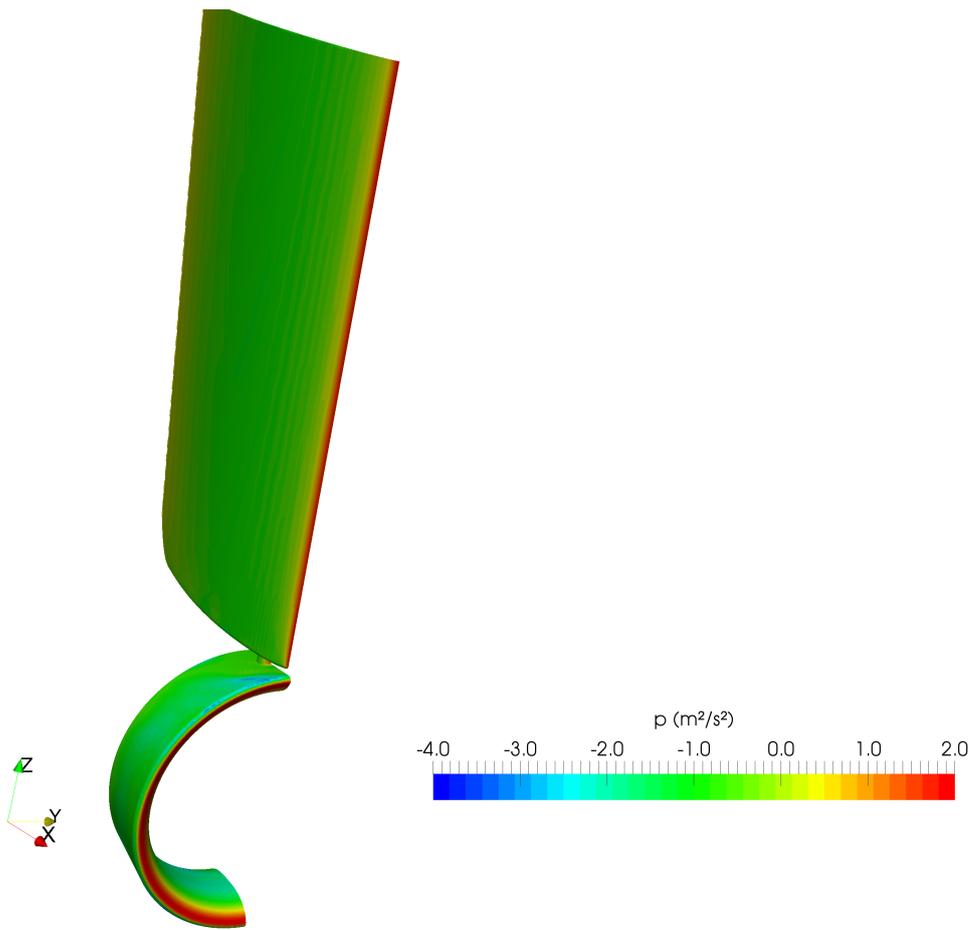


Abbildung B.8: Druck auf Oberfläche Außenseite - mit Halter - Iteration 800

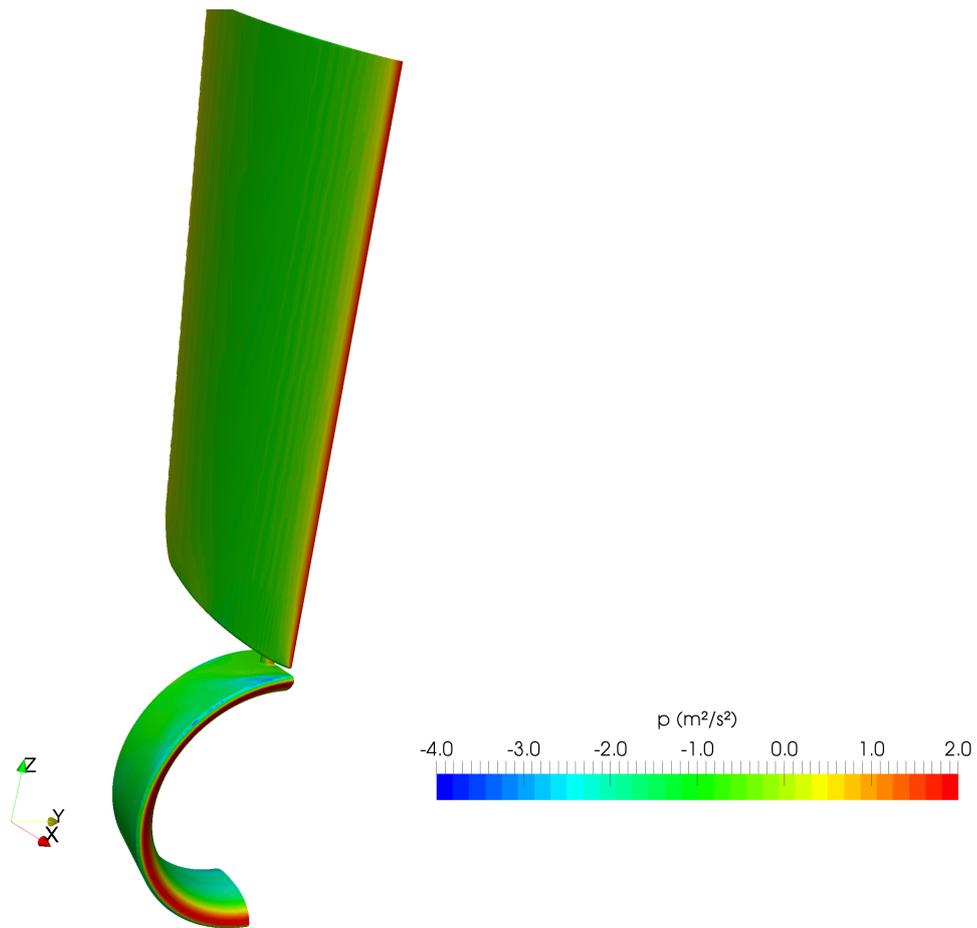


Abbildung B.9: Druck auf Oberfläche Außenseite - mit Halter - Iteration 900

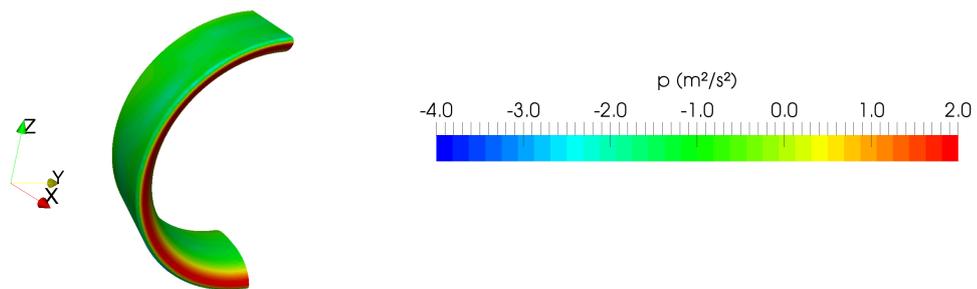


Abbildung B.10: Druck auf Oberfläche Außenseite - ohne Halter - Iteration 800

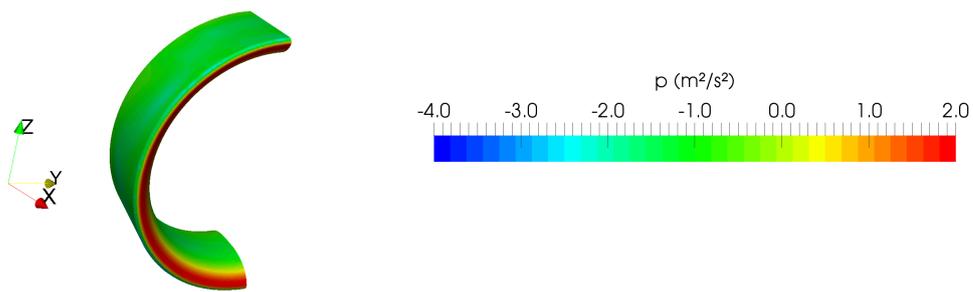


Abbildung B.11: Druck auf Oberfläche Außenseite - ohne Halter - Iteration 900

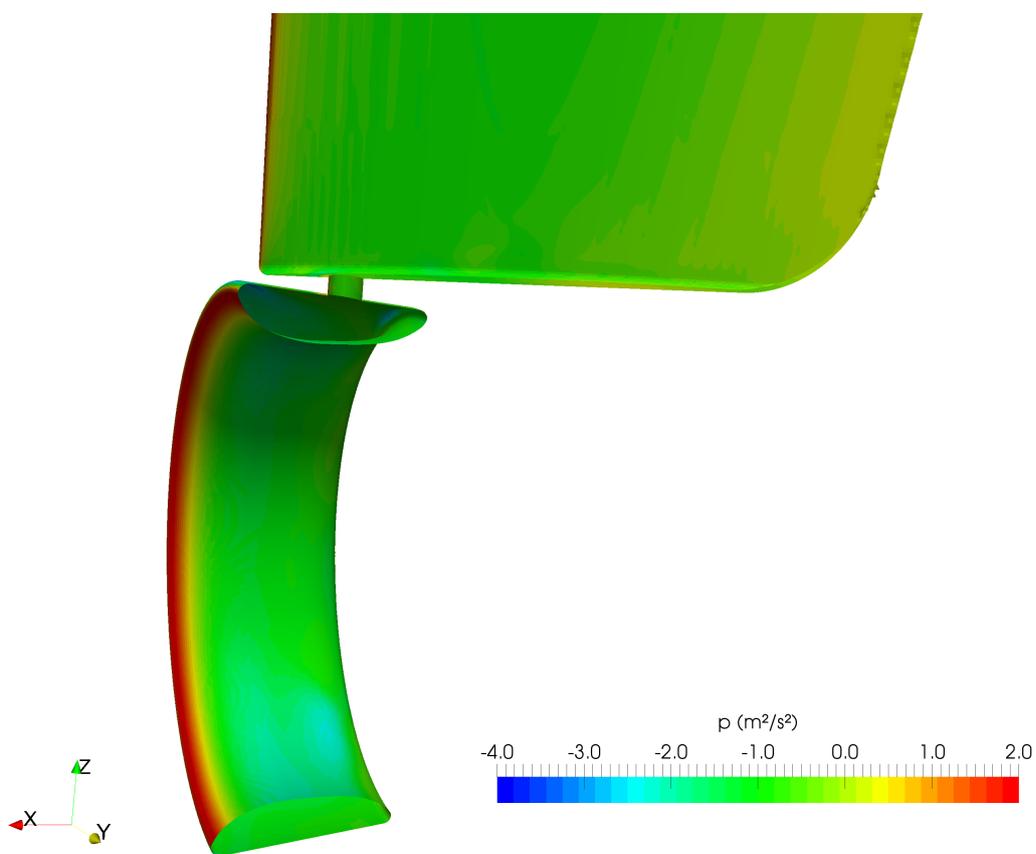


Abbildung B.12: Druck auf Oberfläche Innenseite - mit Halter - Iteration 800

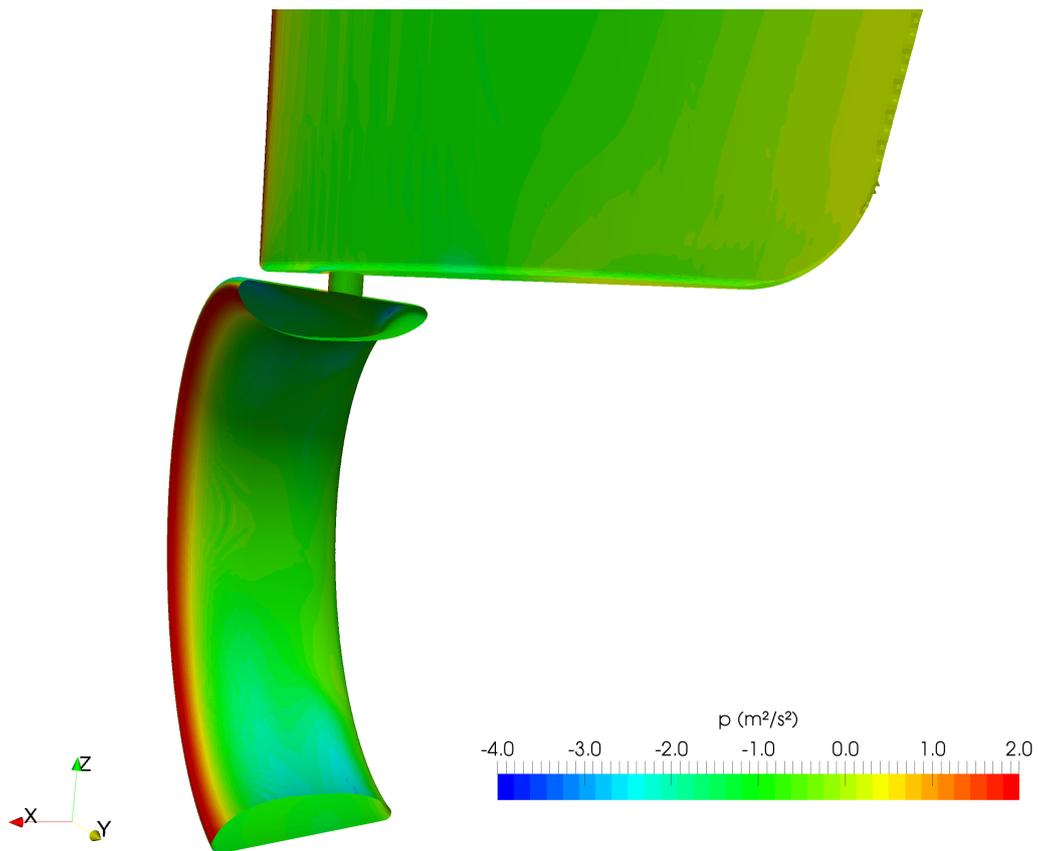


Abbildung B.13: Druck auf Oberfläche Innenseite - mit Halter - Iteration 900

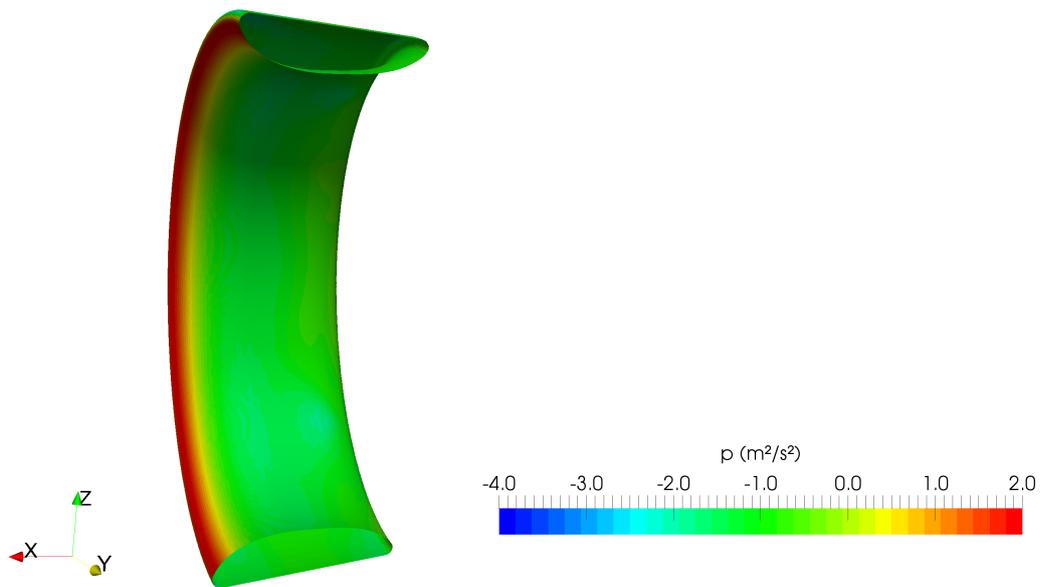


Abbildung B.14: Druck auf Oberfläche Innenseite - ohne Halter - Iteration 800

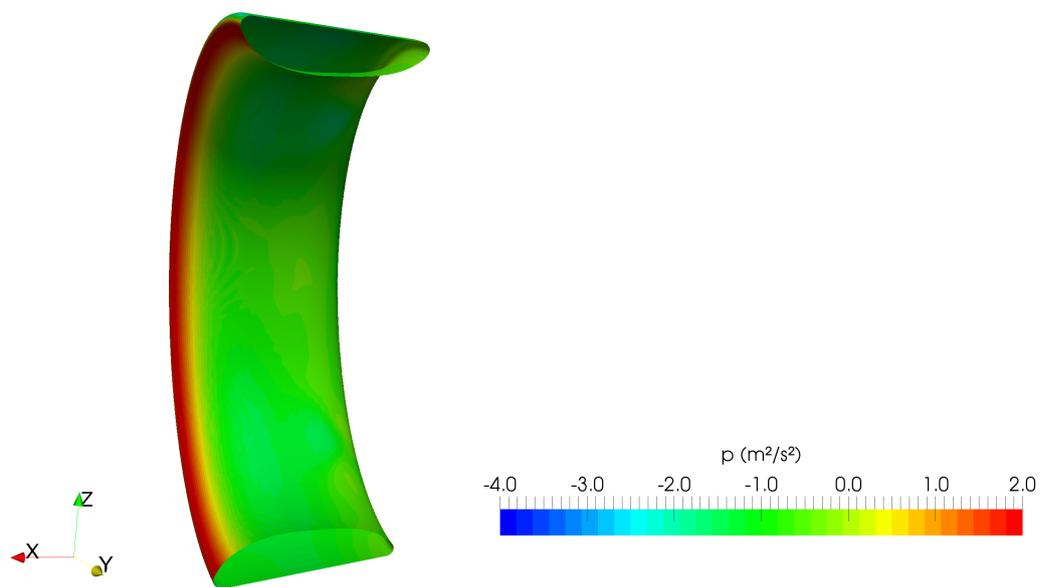


Abbildung B.15: Druck auf Oberfläche Innenseite - ohne Halter - Iteration 900

C Weitere Informationen

C.1 Dateien für die actuatorDisk

C.1.1 topoSetDict

Diese Datei befindet sich im `system` Ordner

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / Operation | Version: 4.0 |
| \\ / And | Web: www.OpenFOAM.org |
| \\ / Manipulation |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  class dictionary;
  object topoSetDict;
}

// * * * * *

actions
(
  // actuationDisk1
  {
    name actuationDisk1CellSet;
    type cellSet;
    action new;
    source cylinderToCell;
    sourceInfo
    {
      p1 (-0.01 0 0);
      p2 (0.01 0 0);
      radius 0.1075;
    }
  }
)

```

```

{
name    actuationDisk1;
type    cellZoneSet;
action  new;
source  setToCellZone;
sourceInfo
{
set actuationDisk1CellSet;
}
}

);

// ***** //

```

C.1.2 fvOptions

Diese Datei befindet sich im **constant** Ordner

```

/*-----*- C++ -*-----*\
| ===== |
| \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / O p e r a t i o n | Version: 4.0 |
| \ \ / A n d | Web: www.OpenFOAM.org |
| \ \ / M a n i p u l a t i o n | |
\*-----*-*/

FoamFile
{
version    2.0;
format     ascii;
class      dictionary;
location   "constant";
object     fvOptions;
}

// * * * * * //

disk1
{
type       actuationDiskSource;
active     on;

actuationDiskSourceCoeffs
{
fields     (U);
}
}

```

```
selectionMode    cellSet;
cellSet          actuationDisk1;
diskDir          (1 0 0);    // Orientation of the disk
Cp              0.3273;
Ct              0.2728;
diskArea        0.036;
upstreamPoint   (-1 0 0);
}
}
```

```
// ***** //
```




Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Backhaus

Vorname: Daniela

dass ich die vorliegende Bachelorarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Untersuchung des hydrodynamischen Einflusses eines Haltearms auf ein Freifahrtgerät anhand von CFD

ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Hamburg

Ort

Datum

Unterschrift im Original