

TU Berlin  
Fakultät V  
Institut für Land- und Seeverkehr  
FG Dynamik maritimer Systeme

## Bachelorarbeit

# Validierung und Verifikation von RANSE-Berechnungen eines Containerschiffes mit OpenFOAM

zugehöriges Modul: Schiffshydrodynamik I  
Erstkorrektor: Prof. Dr. Andrés Cura  
Zweitkorrektor: Dr.-Ing Gonzalo Tampier  
Abgabe: November 2010

von  
Antonio Lengwinat MatrNr.: 315349



# Eidesstattliche Erklärung

Die selbständige und eigenhändige Ausfertigung versichert an Eides statt.

Berlin, den 9. Februar 2011

---

ANTONIO LENGWINAT

## Beschreibung der Aufgabe

Ziel der Arbeit ist eine ausführliche Validierung und Verifikation (wie von ITTC - Uncertainty Analysis in CFD vorgeschlagen) des Glattwasserwiderstandes vom KRISO-Containerschiff, der in wiederholten CFD-Workshops (insb. Gothenburg 2000) untersucht wurde. Sowohl globale (Widerstandsbeiwerte und Formfaktor) als auch lokale Größen (Nachstromfeld und Wellenfeld) sollen mit den Experimenten validiert werden. Zusätzlich sollen die erzielten Ergebnisse mit den von den Workshop-Teilnehmern veröffentlichten Ergebnissen verglichen und kritisch diskutiert werden.

## Kurzfassung

Mit der frei verfügbaren CFD-Software OpenFOAM wurde eine RANSE Berechnung der Umströmung eines Containerschiffs (KCS) in Modellgröße durchgeführt. Die Berechnungen wurden mit und ohne freie Oberfläche durchgeführt, wobei für jeden Fall drei Gitter verwendet wurden, die systematisch verfeinert sind. Die Gittererstellung erfolgte mit dem OpenFOAM Programm `snappyHexMesh`. Die ausgewerteten Größen der Simulation mit freier Oberfläche ( $k$ ,  $C_V$ ,  $V1$  im Nachstrom) und ohne freie Oberfläche ( $C_T$ ,  $C_R$ , Wellenhöhe  $\frac{z}{L}$ ) wurden anschließend einer Verifikation und Validierung nach Empfehlungen der ITTC unterzogen. Als Referenzdaten wurden hierfür die experimentellen Ergebnisse der CFD Workshops Gothenburg 2000 verwendet. Dabei kann der Gesamtwiderstandsbeiwert, mit einer von der ITTC empfohlenen Korrektur, bei knapp 10%  $C_{T_{Data}}$  Abweichung validiert werden, sowie große Teile des Nachstroms. Das Wellenbild und der Nachstrom wurden nicht der Validierungsprozedur unterzogen, diese passen aber sehr gut zu den experimentellen Ergebnissen. Im Vergleich zu den Teilnehmern des Workshops liegen die Ergebnisse von  $C_T$  und  $C_R$  im unteren Mittelfeld, das Wellenfeld sowie der Nachstrom werden jedoch sehr gut wieder gegeben.

## Abstract

The free CFD-Software OpenFOAM was used for RANSE calculation of the flow around a container vessel (KCS) in model size. The calculations were done with and without a free surface, whereat in each case were used three systematically refined grids. The grids were generated with the OpenFOAM program `snappyHexMesh`. With the evaluated variables of the simulation with free surface ( $k$ ,  $C_V$ ,  $V1$  in the wake) and without free surface ( $C_T$ ,  $C_R$ , waveheight  $\frac{z}{L}$ ), a verification and validation was conducted, using the recommendations of the ITTC guidelines. The reference data used in the validation process comes from experimental data generated in the Gothenburg 2000 CFD-Workshop. As a result, the total resistance coefficient  $C_T$  can be validated at 10%  $C_{T_{Data}}$ , after a correction of the result proposed by the ITTC. Mayor parts of the wake velocity  $V1$  can be validated as well. The wave field and the wake (as a whole) were not subjected to the validation procedure, however these fields look very similar to the experimental data. In comparison to the results of the workshop participants, the results of  $C_T$  and  $C_R$  are located below the average value, although the wake and the wave field are reproduced very well.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>2</b>
2.1	Erhaltungsgleichungen . . . . .	2
2.2	Grundlagen numerischer Lösungsmethoden . . . . .	3
2.3	Numerische Diskretisierungsmethoden . . . . .	4
2.4	Lösen der entstandenen Gleichungssysteme . . . . .	5
2.5	Turbulenzmodellierung . . . . .	6
2.6	Modellierung der freien Oberfläche . . . . .	7
2.7	Experimentelle Ermittlung von Schiffswiderständen . . . . .	8
<b>3</b>	<b>Erstellung des numerischen Gitters</b>	<b>9</b>
3.1	Gitter mit freier Oberfläche . . . . .	12
3.2	Gitter ohne freie Oberfläche . . . . .	13
<b>4</b>	<b>CFD mit OpenFOAM 1.6</b>	<b>15</b>
4.1	Dateistruktur von OpenFOAM 1.6 . . . . .	15
4.2	Einstellungen: Simulation ohne freie Oberfläche . . . . .	16
4.3	Einstellungen: Simulation mit freier Oberfläche . . . . .	18
4.4	Berechnen der Druck- und Reibungskräfte . . . . .	19
4.5	Aufgetretene Probleme bei der Simulation . . . . .	20
<b>5</b>	<b>Simulationsergebnisse</b>	<b>21</b>
5.1	Rechnungen ohne freie Flüssigkeitsoberfläche . . . . .	21
5.2	Rechnungen mit freier Flüssigkeitsoberfläche . . . . .	24

<b>6</b>	<b>Verifikation und Validierung</b>	<b>27</b>
6.1	Verifikationsverfahren . . . . .	28
6.2	Validierungsverfahren . . . . .	31
6.3	Verifikation und Validierung - Gitter ohne FO . . . . .	31
6.4	Verifikation und Validierung - Gitter mit FO . . . . .	35
<b>7</b>	<b>Vergleich der Ergebnisse mit anderen Arbeiten</b>	<b>39</b>
<b>8</b>	<b>Abschlussbetrachtung</b>	<b>43</b>
<b>9</b>	<b>Zusammenfassung</b>	<b>44</b>
<b>A</b>	<b>Daten des KRISO Containerschiffs</b>	<b>A1</b>
<b>B</b>	<b>Gitterparameter</b>	<b>A2</b>
<b>C</b>	<b>Rechnungen und Graphen des Validierungsverfahrens</b>	<b>A4</b>
C.1	Gitter ohne FO . . . . .	A4
C.2	Gitter mit FO . . . . .	A9
C.3	Ergebnisse von Gothenburg 2000 . . . . .	A10
<b>D</b>	<b>Genutzte Funktionen von Linux und OpenFOAM</b>	<b>A10</b>

# Tabellenverzeichnis

1	Abkürzungsverzeichnis . . . . .	ix
2	Symbolverzeichnis . . . . .	xi
3	allgemeine Gitterparameter . . . . .	10
4	Zelldaten der Gitter mit FO . . . . .	12
5	Zelldaten der Gitter ohne FO . . . . .	13
6	Lösungsschemen der Simulation ohne FO . . . . .	17
7	Lösungsalgorithmen der Simulation ohne FO . . . . .	18
8	Lösungsschemen der Simulation mit FO . . . . .	18
9	Lösungsalgorithmen der Simulation mit FO . . . . .	19
10	Ergebnisse von $k$ und $C_V$ für Gitter ohne FO . . . . .	21
11	Ergebnisse von $C_T$ und $C_R$ für Gitter mit FO . . . . .	24
12	Hauptabmessungen KCS . . . . .	A1
13	Experimentelle Daten . . . . .	A1
14	Randbedingungen der Simulation ohne FO . . . . .	A2
15	Randbedingungen der Simulation mit FO . . . . .	A3
16	Zusammenfassung der Validierungsergebnisse . . . . .	A4
17	Auswertung von V1 - Teil 1 . . . . .	A6
18	Auswertung von V1 - Teil 2 . . . . .	A7
19	Gothenburg 2000, Ergebnisse der Teilnehmer . . . . .	A10

# Abbildungsverzeichnis

1	Ansicht der Modellgeometrie des KCS . . . . .	9
2	Layer im feinen Gitter mit FO: vertikaler Schnitt bei $x=0$ kl. Bild: Heckansicht nach <code>snappyHexMesh</code> . . . . .	11
3	mittleres Gitter mit FO . . . . .	13
4	mittleres Gitter ohne FO . . . . .	14
5	Nachstromfeld ohne FO, links Gitter 3, rechts Gitter 2 . . . . .	22
6	Nachstromfeld ohne FO, links Gitter 1, rechts EFD (Experiment) . . . . .	23
7	axialer Geschwindigkeitsschnitt bei $x/L=0,4911$ und $z/L=-0,03$ . . . . .	23
8	Wellenfeld, oben Gitter 3, unten Gitter 2; $waveH$ in m . . . . .	25
9	Wellenfeld, oben Gitter 1, unten EFD; $waveH$ in m . . . . .	26
10	Wellenhöhe $z/L$ im Längsschnitt bei $y/L=0,1509$ . . . . .	26
11	Wellenfeld des KCS mit Neptun-Software (Gothenburg 2000) <i>Quelle: Archivdaten von Prof. Dr. Andrés Cura</i> . . . . .	42
12	yPlus des feinen Gitters ohne FO . . . . .	A3
13	Iterative Unsicherheit von $C_V$ beim Gitter ohne FO . . . . .	A4
14	Konvergenzradius von $V1$ in jedem Punkt . . . . .	A5
15	Abweichung des berechneten $V1$ vom Experiment . . . . .	A5
16	Vergleich - Unsicherheiten und Fehler von $V1$ . . . . .	A8
17	Vergleich - korrigierte Unsicherheiten und Fehler von $V1$ . . . . .	A8
18	Schwankung von $C_T$ des feinen Gitters nach 40000 Iterationen . . . . .	A9
19	punktuelter Konvergenzradius der Wellenhöhe $z/L$ im Wellenlängsschnitt ( $y/L=0,1509$ ) . . . . .	A9

# Abkürzungsverzeichnis

<b>Abkürzung</b>	<b>Bedeutung</b>
CFD	Computational Fluid Dynamics
DIC	Diagonal incomplete-Cholesky
DNS	Direct Numerical Simulation
EFD	Experimental Fluid Dynamics
FD	Finite Differenzen (Methode)
FO	freie Flüssigkeitsoberfläche
FV	Finite Volumen (Methode)
GAMG	Geometric-algebraic multi-grid
GS	Gleichungssystem
ITTC	International Towing Tank Conference
KCS	KRISO Container Ship
KV	Kontrollvolumen
KRISO	Korean Research Institute for Ships and Ocean Engineering
LES	Large Eddy Simulation
OpenFOAM	Open Field Operation and Manipulation
PISO	Pressure-implicit Split-operator
RANSE	Reynolds averaged navier stokes equations
RE	Generalisierte Richardson Extrapolation
SIMPLE	Semi-implicit Method for Pressure-linked Equations
SST	shear stress transport
UA	Uncertainty Analysis
VOF	Volume of Fluid

**Tabelle 1** Abkürzungsverzeichnis

# Symbolverzeichnis

Symbol	Bedeutung	Einheit
$a$	Beschleunigung	$\frac{m}{s^2}$
$b$	Feldkräfte pro Masse (engl. Body-forces)	$\frac{m}{s^2}$
$C_F$	Reibungswiderstandsbeiwert	1
$C_{ITTC}$	Reibungswiderstandsbeiwert nach ITTC Formel	1
$C_k$	Korrekturfaktor des abgeschätzten Fehlers	1
$C_{PV}$	viskoser Druckwiderstandsbeiwert	1
$C_R$	Restwiderstandsbeiwert	1
$C_T$	Gesamtwiderstandsbeiwert	1
$C_V$	viskoser Widerstandsbeiwert	1
$D$	Experimenteller Datenwert	variiert
$E$	Vergleichsfehler	variiert
$F_S$	Sicherheitsfaktor in der Verifikation	1
$k$	kinetische Energie der Turbulenz	$\frac{m^2}{s^2}$
$k$	Formfaktor des Schiffsrumpfs	1
$L$	Schiffslänge = $L_{PP}$	$m$
$L_{PP}$	Länge zwischen den Loten	$m$
$P_k$	Genauigkeitsgrad der Fehlerabschätzung	$m$
$p$	Druck	$\frac{N}{m^2}$
$R_F$	Reibungswiderstand	$N$
$R_k$	Konvergenzradius einer Variable	1
$Rn$	Reynoldszahl	1
$R_P$	Druckwiderstand	$N$
$R_G$	Konvergenzradius des Gitters	1
$r_k$	Verfeinerungsfaktor zwischen Gittern	1
$S$	Simulationsergebnis einer Variable	variiert
$S_C$	korrigiertes Simulationsergebnis einer Variable	variiert
$S_0$	benetzte Fläche	$m^2$
$t$	Zeit	$s$
$T$	Realwert (engl. truth)	variiert
$U$	Geschwindigkeit	$\frac{m}{s}$
$U_D$	Unsicherheit der experimentellen Daten	variiert
$U_G$	Gitterunsicherheit	variiert
$U_I$	iterative Unsicherheit	variiert
$U_k$	Unsicherheit einer Variable k	variiert
$U_P$	Unsicherheit anderer Parameter	variiert
$U_{SN}$	numerische Unsicherheit der Simulation	variiert
$U_{S_C N}$	korrigierte numerische Unsicherheit der Simulation	variiert
$U_T$	Unsicherheit im Zeitschritt	variiert
$U_V$	Validierungsunsicherheit	variiert
$V1$	normierte Geschwindigkeit bei $\frac{x}{L} = 0,4911$ und $\frac{z}{L} = -0,03$	1

Symbol	Bedeutung	Einheit
$\alpha_1$	Konzentration der Phase 1 in der Zelle	1
$\delta$	Fehlergröße	variiert
$\delta^*$	abgeschätzte Fehlergröße	variiert
$\delta_D$	Fehler im experimentellen Datenwert	variiert
$\delta_{SM}$	Modellfehler	variiert
$\delta_{SN}$	numerischer Fehler	variiert
$\delta_{RE}^*$	abgeschätzter Fehler per Richardson Extrapolation	variiert
$\epsilon$	turbulente Dissipation	$\frac{m^2}{s^3}$
$\varepsilon$	Fehler in der Fehlerabschätzung	variiert
$\varepsilon_{k_{ij}}$	Änderung zwischen zwei Lösungen	variiert
$\lambda$	Modellmaßstab	1
$\mu$	dynamische Viskosität	$\frac{N \cdot s}{m^2}$
$\nu$	kinematische Viskosität	$\frac{m^2}{s}$
$\nu_t$	Wirbelviskosität im Turbulenzmodell	$\frac{m^2}{s}$
$\omega$	turbulente Frequenz	$\frac{1}{s}$
$\rho$	Dichte	$\frac{kg}{m^3}$
$\tau$	Schubspannung	$\frac{N}{m^2}$

Tabelle 2 Symbolverzeichnis

# 1 Einleitung

Seit William Froude im Jahre 1867 mit den Untersuchungen an Schiffsmodellen begann, hat sich einiges geändert. Es wurden zusätzliche Verfahren entwickelt, um bestimmte Kennzahlen zu ermitteln, Reibungslinien wurden verbessert und alle wichtigen Vorgänge vom internationalen Zusammenschluss der Versuchsanstalten (ITTC) genormt. Das ursprüngliche Problem aber, dass bei Modellversuchen nur die Froude'sche Ähnlichkeit und nicht gleichzeitig die Reynolds'sche eingehalten werden kann und die daraus resultierenden Probleme bleiben bestehen. Bei der Widerstandsanalyse kann so z.B. nur der Restwiderstandsbeiwert  $C_R$  und der Formfaktor  $k$  direkt auf die Großausführung übertragen werden. Der Restwiderstandsbeiwert beträgt in vielen Fällen aber nur ca. 10% des Gesamtwiderstandsbeiwertes  $C_T$ .

Mit den Möglichkeiten von CFD Rechnungen könnte dieses Problem nun umgangen werden. Ideal wäre es sicherlich, ein Schiff nur mit Hilfe von CFD Rechnungen auszulegen. Davon sind wir, wegen den existenten Rechenkapazitäten großer Rechner, immer noch weit entfernt. Momentan werden per CFD nur Strukturen in Modellgröße untersucht, um letztendlich auch nur die direkt übertragbaren Größen zu ermitteln. Man erhält aber auch gleichzeitig das Nachstrombild, sowie das Wellenbild bei Rechnungen mit freier Oberfläche.

Über die letzten zwei Jahrzehnte wurde einiges an CFD-Software entwickelt, das für Schiffsrumpfuntersuchungen geeignet ist. Größtenteils wurde sie von anwendenden Instituten und Firmen selbst geschrieben oder wird teuer verkauft. Im Jahr 2004 kam erstmals die kostenlose CFD-Software OpenFOAM auf den Markt. Bis heute verfügt sie über wenig und grob gehaltene Dokumentation. Die Software an sich wird aber kontinuierlich erweitert und bietet vielseitige Möglichkeiten für die Simulation von Schiffsumströmungen.

Mit CFD-Simulationen ist es aber auch möglich komplett falsche oder "aus der Luft gegriffene" Ergebnisse zu produzieren. Um dem entgegen zu wirken und CFD-Berechnungen allgemein mehr Anerkennung zu verleihen, wurden verschiedene Validierungsprozeduren erdacht. Die ITTC veröffentlichte 2002 eine Reihe von Richtlinien, nach denen eine solide Validierung durchzuführen ist [2].

In der vorliegenden Arbeit wird mit OpenFOAM ein moderner Containerschiffsrumpf (das KRISO Container Ship) untersucht, der auch in einigen CFD Workshops wie Gothenburg 2000 und Tokyo 2005 betrachtet wurde. Die Arbeit ist beschränkt auf die grundlegende Untersuchung der gleichförmigen Bewegung des Schiffes unter Glattwasserbedingungen. Die anschließende Verifikation und Validierung der erhaltenen Ergebnisse wurden nach ITTC Standard durchgeführt. Dabei ist vor allem interessant, ob die Ergebnisse der drei erstellten Gitter gegen einen gitterunabhängigen Wert konvergieren oder nicht.

## 2 Theoretische Grundlagen

Vor der Durchführung numerischer Berechnungen an Strömungen sollten die grundlegenden Gleichungen der Strömungslehre sowie die damit verbundenen Phänomene bekannt sein. In diesem Abschnitt wird kurz auf die notwendigen Gleichungen und getroffenen Annahmen eingegangen.

Im Folgenden werden alle Gleichungen (sofern möglich) der Übersichtlichkeit halber in Indexschreibweise wiedergegeben. Alle nachfolgend beschriebenen und somit für die durchgeführten Rechnungen verwendeten Formeln, gehen von Inkompressibilität aus. Dies ist der Fall, da Wasser und Luft bei den auftretenden Geschwindigkeiten und Drücken als inkompressibel modelliert werden können, ohne einen bedeutenden Fehler einzuführen. Bei Luft muss erst ab einer Machzahl größer 0,3 Kompressibilität angenommen werden. [9] Bei den untersuchten Medien handelt es sich um Newton-Fluide, daher können Spannungen und somit Kräfte am untersuchten Schiffskörper mit dem Newton'schen Schubspannungsansatz berechnet werden.

$$\tau = \mu \frac{du}{dy} \quad (1)$$

Die numerische Strömungsmechanik stützt sich auf die Sichtweise, Fluide als Kontinua anzunehmen. Dabei ist ein Kontinuum ein Stoff, der das betrachtete Gebiet (Kontrollvolumen) vollständig ausfüllt und somit alle Stoffeigenschaften auch in einer noch so kleinen Umgebung erhalten bleiben. [9] Daher geht die molekulare Struktur nur über Transportvorgänge auf molekularer Ebene, wie Wärmetransport und Reibung ein. Diese Transporte werden Diffusionsvorgänge genannt.

Für die Beschreibung einer Strömung gibt es zwei grundlegende Betrachtungsweisen. Einmal die teilchenfeste Betrachtung (Lagrange-Koordinaten), bei der die Koordinaten von einer Referenzlage abhängen ( $\vec{x} = [x(x_0, y_0, z_0, t), y(y_0, \dots)]$ ), die Geschwindigkeiten aber absolute Werte sind ( $\vec{u} = (u_x, u_y, u_z)$ ). Zum anderen die raumfeste Betrachtung (Euler-Koordinaten), bei der die Raumkoordinaten absolut sind, die Geschwindigkeiten aber von der Position im Raum und der Zeit abhängen ( $\vec{u} = [u_x(x, y, z, t), u_y(x, y, z, t), u_z(x, y, z, t)]$ ). [11] Die letztere hat sich in der Strömungsmechanik weitestgehend durchgesetzt und wird auch in dieser Arbeit verwendet.

### 2.1 Erhaltungsgleichungen

Um die gewünschten Größen einer Strömung (z.B.  $u_x, u_y, u_z, p$ ) ermitteln zu können, muss ein System aus genauso vielen Gleichungen wie Unbekannten aufgestellt werden. Diese Gleichungen sind Erhaltungsgleichungen oder basieren auf diesen. Alle Erhaltungsgesetze sind herleitbar über die Betrachtung der Masse, des Impulses und der Energie in einem Kontrollvolumen. [5]

Eine Sonderstellung nimmt das **Reynolds Transporttheorem** (Gl. (2)) ein, da mit dessen Hilfe die anderen Erhaltungsgleichungen hergeleitet werden können. Die Größe  $\phi$  steht dabei als Platzhalter für eine intensive Eigenschaft des Kontinuums geteilt durch die Dichte. Somit ergibt sich zum Beispiel die Massenerhaltung bei Einsetzen der Zahl 1 und die Impulserhaltung bei Einsetzen der Geschwindigkeit  $\vec{u}$  anstelle von  $\phi$ . [5]

$$\frac{d}{dt} \int_{V_{KM}} \rho \phi \, dV = \frac{d}{dt} \int_{V_{KV}} \rho \phi \, dV + \int_{S_{KV}} \rho \phi (\vec{u} - \vec{u}_b) \cdot \vec{n} \, dS \quad (2)$$

Die **Massenerhaltung** für ein inkompressibles Fluid sagt aus, dass die Summe der Geschwindigkeitsänderungen in allen Koordinatenrichtungen (x,y,z) gleich Null ist. Die Gleichung kann in verschiedenen Formen dargestellt werden, Gleichung (3) ist die Differentialform.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \quad \xrightarrow{\text{inkompressibel}} \quad \frac{\partial u_i}{\partial x_i} = 0 \quad (3)$$

Die **Impulserhaltung** basiert auf dem zweiten Newton'schen Gesetz  $\vec{F} = m \cdot \vec{a}$ . Der Impuls bleibt in jeder kartesischen Koordinatenrichtung erhalten, wodurch drei linear unabhängige Gleichungen entstehen, die in Gleichung (4) enthalten sind.

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \rho b_i \quad (4)$$

Die **Energieerhaltung** entspringt dem 1. Hauptsatz der Thermodynamik. Sie kann in unterschiedlichen Formen dargestellt werden, je nachdem welche Größe betrachtet werden soll. Die betrachteten Vorgänge bei Schiffsströmungsberechnungen sind fast immer isotherm, daher wird diese Gleichung nicht berücksichtigt und hier nicht aufgeführt.

## 2.2 Grundlagen numerischer Lösungsmethoden

Der Aufbau numerischer Software für die Lösung von Strömungsproblemen ist grundlegend immer der gleiche. Als erstes muss ein *mathematisches Modell* erstellt werden. Dies ist ein System aus nichtlinearen partiellen Differentialgleichungen, die möglicherweise die Realität vereinfachen, inklusive der nötigen Randbedingungen für die Lösungsgenerierung. Anschließend muss eine passende Diskretisierungsmethode für das Problem gewählt werden, um die Differentialgleichungen durch ein System von algebraischen Gleichungen zu ersetzen. In dieser Arbeit wird das Softwarepaket OpenFOAM genutzt und somit ist die Finite-Volumen-Methode (FV) verwendet worden. Andere Optionen sind Diskretisierungen per Finite-Differenzen- oder Finite-Elemente-Methode.

Ein numerisches Gitter ist notwendig, um das Lösungsgebiet auf eine endliche Anzahl von Punkten zu beschränken. Dabei gibt es viele verschiedene Arten von Gittern. Die einfachsten sind *strukturierte Gitter*. Sie bestehen aus Sätzen von Gitterlinien, wobei sich die Linien eines Satzes nicht begegnen und jedes Mitglied eines anderen Satzes nur einmal schneiden. *Blockstrukturierte Gitter* sind etwas komplexer. Sie bestehen aus mehrstufigen Unterteilungen des Lösungsgebietes. Jeder Block beinhaltet ein strukturiertes Gitter und stellt einen Teil des Lösungsgebietes dar. Auf diese Weise können komplexere Strömungsgebiete aufgelöst werden. *Unstrukturierte Gitter* sind am flexibelsten und können sich an beliebige Geometrien anpassen. Ausführlich erklärt werden die Gittertypen mit ihren Eigenheiten im Buch von Ferziger & Peric [5].

Nach der Diskretisierung per FV muss das entstandene Gleichungssystem (GS) für jeden Punkt des Gitters mit einer geeigneten Lösungsmethode gelöst werden. Da es sich bei den mit CFD untersuchten Strömungen meistens um sehr komplexe Vorgänge handelt, können die nichtlinearen Gleichungssysteme unter vertretbarem Rechenaufwand nur iterativ gelöst werden. Dadurch wird das GS sukzessive linearisiert, die Wahl des Löser ist dem Nutzer überlassen. Sie hängt unter anderem vom Gittertyp ab.

Numerische Lösungsmethoden müssen bestimmte Voraussetzungen erfüllen. Als erstes muss eine numerische Lösung konsistent sein. Dafür ist der Abbruchfehler wichtig, er ist der Unterschied zwischen der Lösung der exakten Gleichungen und der diskretisierten. Für Konsistenz muss dieser Fehler gegen Null gehen, wenn die Diskretisierung unendlich fein wird ( $\Delta t \rightarrow 0$  und  $\Delta x \rightarrow 0$ ).

Die Lösung muss stabil sein. Das bedeutet, das Verfahren darf kleine Fehler in den Daten nicht anfachen. Insbesondere dürfen Rundungsfehler dabei das Ergebnis nur minimal beeinflussen.

Die Konvergenz einer Lösungsmethode ist wahrscheinlich ihre wichtigste Eigenschaft. Konvergenz ist erfüllt, wenn die Lösung des diskretisierten Systems mit gegen Null strebenden Gitterabständen gegen die reale Lösung der Differentialgleichungen strebt.

Am Ende ist es wichtig die Genauigkeit der Lösung zu kennen, da es sich nur um eine Näherung der Realität handelt. Dabei wird der Fehler in der Lösung abgeschätzt, der auf jeden Fall drei Arten von Fehlern beinhaltet: Modellfehler, Diskretisierungsfehler und Iterationsfehler [5]. Zusätzlich zu diesen gibt es Phänomene wie die numerische Diffusion. Sie entsteht, wenn die berechneten Stromlinien einen großen Winkel zu den Gitterlinien haben, wodurch beispielsweise die Grenzschicht dicker dargestellt wird als real vorhanden.

## 2.3 Numerische Diskretisierungsmethoden

Für die Diskretisierung der Gleichungen gibt es drei verschiedene Methoden.

Die **Finite Differenzen Methode** (FD) ist die Annäherung der Differentiale per Differenzen, Ausgangspunkt sind die Erhaltungsgleichungen in Differentialform. Falls gewünscht,

kann diese Annäherung sehr genau durchgeführt werden. FD stellt gleichzeitig aber große Ansprüche an die Netzqualität. Dadurch sind nur Rechnungen auf strukturierten Gittern möglich. Das macht die Methode vergleichsweise unflexibel und nicht brauchbar für sehr komplexe Anwendungsgebiete, wie eine Schiffsumströmung.

Die **Finite-Volumen Methode** (FV) wird heutzutage am meisten verwendet und ist auch in OpenFOAM umgesetzt. Der Ausgangspunkt der FV sind die Erhaltungsgleichungen in Integralform. Das Lösungsgebiet wird in eine endliche Anzahl an Kontrollvolumina (KV) unterteilt, welche jeweils einen Rechenpunkt in ihrem Schwerpunkt besitzen. Für jedes KV müssen die Erhaltungsgleichungen erfüllt sein. Dadurch kann diese Methode mit jedem Gittertyp verwendet werden. Der Nachteil ist, dass Verfahren höherer Ordnung (als 2.) hierbei schwer umzusetzen sind und man dadurch einen Genauigkeitsverlust in Kauf nimmt.

Die **Finite Elemente Methode** basiert auf den Gleichungen in Integralform und wird generell bei unstrukturierten Gittern verwendet. Der Unterschied zu FV ist, dass die Gleichungen mit einer Gewichtungsfunktion multipliziert werden. Die Methode wird hauptsächlich bei Festkörperberechnungen eingesetzt.

## 2.4 Lösen der entstandenen Gleichungssysteme

Nach einer geeigneten Diskretisierung müssen die entstandenen linearen oder nichtlinearen Gleichungssysteme der Form  $A \cdot \phi = \mathbf{Q}$  gelöst werden. Lineare Systeme können teilweise mit direkten Verfahren wie der LU-Zerlegung gelöst werden.

Die Navier-Stokes-Gleichungen sind nichtlinear und erfordern daher einen anderen Lösungsansatz. Für komplexe lineare und nichtlineare Probleme greift man auf iterative Lösungstechniken zurück. Diese schätzen die Gleichungen ab, linearisieren und korrigieren die Lösung. Dies ist ein Wiederholungsprozess, der bis zur Konvergenz des Ergebnisses durchgeführt wird. Der Rechenaufwand wird dabei in einem akzeptablen Rahmen gehalten und das Gleichungssystem ausreichend genau gelöst.

In den folgenden Kapiteln wird unterschieden zwischen Lösungsprogrammen, Lösungsschemen und Lösungsalgorithmen. Lösungsprogramme sind in OpenFOAM per Befehl aufrufbare Programme, die einen Simulationsfall lösen können. Sie greifen dabei auf festgelegte Lösungsschemen und Lösungsalgorithmen zurück, um die Gleichungssysteme zu lösen. Die Programme geben in definierten Abständen Ergebnisse der Rechnung aus und können bei Bedarf jederzeit beendet werden.

Lösungsalgorithmen sind mathematische Algorithmen zur iterativen Lösung nichtlinearer Gleichungssysteme. Sie greifen auf die entstandenen Ausdrücke nach Anwendung der Lösungsschemen zurück und lösen (annähernd) das entstandene Gleichungssystem.

Lösungsschemen legen fest, wie die Ausdrücke (Ableitungen) in den Gleichungen angenähert bzw. linearisiert werden. Vom genutzten Verfahren hängt auch die Genauigkeit der Lösung ab.

## 2.5 Turbulenzmodellierung

Schiffsumströmungen sind immer turbulent, daher ist die Modellierung dieses Strömungsphänomens essentiell. An der mathematischen Struktur der Gleichungen ändert sich durch Turbulenz nichts Wesentliches. Die Navier-Stokes Gleichungen beschreiben auch turbulente Strömungen und können somit per direkter numerischer Simulation (**DNS**) gelöst werden. Dabei muss das Gebiet aber so fein aufgelöst sein, dass auch kleinste turbulente Wirbel (Eddie) erfasst werden. Eine Zellenlänge darf daher nicht größer sein als ein Kolmogoroff-Längenmaß [9], was proportional zu  $Re^{-\frac{3}{4}}$  und somit sehr klein ist. Die DNS ist daher nicht effizient in ingenieurtechnischen Anwendungen einsetzbar, hohe Reynoldszahlen können nicht erreicht werden und der Rechenaufwand ist enorm groß.

Die nächst genauere Modellierungsstufe nach DNS ist die Large Eddy Simulation (**LES**). Diese löst Eddies erst ab einer bestimmten Größe auf, was die Ergebnisse etwas ungenauer macht, dafür aber Simulation bei höheren Reynoldszahlen erlaubt. Der Rechenaufwand ist für Anwendungen im Ingenieurbereich immer noch viel zu hoch und die Ergebnisse genauer als nötig.

Für die durchgeführten Simulationen in OpenFOAM wurde ein Reynolds Averaged Navier Stokes Modell (**RANS-Modell**) verwendet. Bei RANS geht man davon aus, dass die Strömung statistisch stationär ist, wodurch die Variablen mit einem zeitlich gemittelten Wert und einer Schwankung dargestellt werden können. [5]

$$\phi(x_i, t) = \bar{\phi}(x_i) + \phi'(x_i, t) \quad (5)$$

Diese Vereinfachung kann in die Navier-Stokes Gleichung eingesetzt werden und man erhält die RANS Gleichungen (6) in Tensorschreibweise:

$$\frac{\partial(\rho\bar{u}_i)}{\partial t} + \frac{\partial}{\partial x_j} (\rho\bar{u}_i\bar{u}_j + \overline{\rho u'_i u'_j}) = -\frac{\partial\bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \mu \left( \frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i} \right) \right] \quad (6)$$

Bei der Mittelung der NST-Gleichungen tritt der symmetrische Tensor zweiter Stufe  $\overline{\rho u'_i u'_j}$  mit seinen sechs Unbekannten auf. Die Einträge des Tensors werden *Reynolds-Spannungen* genannt. Durch sie wird das Gleichungssystem unterbestimmt, da nun neue Unbekannte hinzukommen, mehr Gleichungen aber nicht. Es ist erforderlich den Term zu modellieren, was über die Annahme einer erhöhten Viskosität ( $\mu_t =$  Wirbelviskosität) geschieht.

$$-\overline{\rho u'_i u'_j} = \mu_t \left( \frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i} \right) - \frac{2}{3}\rho\delta_{ij}k \quad (7)$$

Die Turbulenz (und somit die Wirbelviskosität) kann über ihre kinetische Energie  $k$  und eine charakteristische Länge  $L$  bestimmt werden. Dies ermöglicht die Berechnung von  $\mu_t$  über Gleichungsmodelle.

*Nullgleichungsmodelle* werden auch Mischungswegmodelle genannt. Sie sind nur für relativ

einfache Strömungen geeignet, da man  $L$  als eine Funktion der Koordinaten vorgeben muss. *Eingleichungsmodelle* lösen eine Transportgleichung für  $k$  und nutzen eine empirische Formel für das Längenmaß.

*Zweigleichungsmodelle* sind die übliche Modellierungsart in Ingenieursanwendungen. Es werden dabei zwei zusätzliche Transportgleichungen gelöst um die Wirbelviskosität an allen Punkten bestimmen zu können. Die drei wichtigsten Modelle sind nachfolgend aufgeführt.

Das **k-epsilon Modell** verwendet zwei Gleichungen für die kinetische Energie der Turbulenz ( $k$ ) und die turbulente Dissipation ( $\epsilon$ ). Das Modell ist gut geeignet für wandferne Strömung, hat aber Probleme bei Strömungen, die sich wegen des Druckgradienten von der Wand ablösen. Das Modell liefert hierbei zu optimistische Prognosen.

Das **k-omega Modell** verwendet  $k$  und die Kunstgröße omega ( $\omega$ ), die als turbulente Frequenz angesehen werden kann. Das Modell liefert auch bei einfacher Auflösung der Grenzschicht gute Ergebnisse in Bezug auf Ablösung. Dafür sind die Simulationsergebnisse im wandfernen Bereich etwas schlechter als beim k-epsilon Modell.

Für die Simulation wurde das **k-omega-SST** Modell (SST = shear stress transport) verwendet. Es kombiniert die vorteilhaften Eigenschaften der beiden genannten Zweigleichungsmodelle mit Hilfe einer "Blending-Funktion". Das Modell ist robust, sowie die Rechenzeiten akzeptabel. Daher wird es in industriellen Anwendungen heute als Standard verwendet. Der entscheidende Zusammenhang im k-omega Modell ist:  $\nu_t = \frac{k}{\omega}$ .

In der Nähe von Wänden muss bei Verwendung eines Zweigleichungsmodells sehr fein aufgelöst werden, um die Grenzschicht richtig zu erfassen. Die feine Auflösung würde den Rechenaufwand aber stark erhöhen, daher greift man auf Wandfunktionen zurück. Diese nutzen zB. das logarithmische Wandgesetz für den Geschwindigkeitsverlauf in der Grenzschicht. Es wird ein dimensionsloser Wandabstand ( $y^+$ ) zwischen Wand und Mittelpunkt der ersten Zelle eingeführt. Der Wandabstand sollte dabei natürlich im logarithmischen Bereich liegen, also  $30 \leq y^+ \leq 200$ .

Die in OpenFOAM enthaltenen Wandfunktionen sind unabhängig vom  $y^+$ -Wert, da sie sich an ihn anpassen und das jeweilige Wandgesetz in Kraft setzen. Im feinen Gitter ohne freie Oberfläche wurde interessehalber der  $y^+$ -Wert ausgegeben (siehe Abb. 12), um zu sehen, wie gleichmäßig der Verlauf ist.

## 2.6 Modellierung der freien Oberfläche

Es gibt verschiedene Techniken, um eine nicht mischbare Zweiphasenströmung und somit eine freie Oberfläche zu simulieren. In OpenFOAM ist die **Volume of Fluid** Methode (VOF) umgesetzt. Diese Technik zur Verfolgung der freien Oberfläche garantiert die Massenerhaltung. Hierfür wird eine zusätzliche Transportgleichung für die Variable alpha1 ( $\alpha_1$ ) gelöst. Die Variable wird in Fachbüchern oft  $C$  genannt, da OpenFOAM 1.6 sie al-

pha1 nennt, wird dies der Konsistenz halber übernommen.  $\alpha_1$  gibt den Volumenanteil an Flüssigkeit an, daher gilt  $0 \leq \alpha_1 \leq 1$ . Bei  $\alpha_1 = 1$  ist die Phase Wasser, bei  $\alpha_1 = 0$  ist sie Gas (in unserem Fall Luft). Die zugehörige Transportgleichung lautet

$$\frac{\partial \alpha_1}{\partial t} + \nabla \cdot \frac{\partial}{\partial x_i} (\alpha_1 u_i) = 0 \quad (8)$$

Wenn  $\alpha_1$  zwischen 0 und 1 liegt, signalisiert dies die freie Oberfläche (FO). Dadurch erhält man keine klare Grenze zwischen den Fluiden, sondern einen Grenzbereich, was die Oberflächenrekonstruktion erschwert.

Eine weitere Art der Erfassung freier Oberflächen ist die **Level-Set** Methode. Es wird hierbei eine Fläche (die freie Oberfläche) und eine "Level-Set" Funktion ( $\phi$ ) definiert. Die Funktion ist gleich Null auf der Fläche.  $\phi$  wird meistens als eine Abstandsfunktion von der Nullstelle definiert. Dabei werden die Phasen durch das Vorzeichen von  $\phi$  identifiziert,  $\phi < 0$  ist Fluid 1 und  $\phi > 0$  Fluid 2. Nun kann diese Funktion  $\phi$  als Lösung der Transportgleichung geschrieben werden.

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \frac{\partial}{\partial x_i} (\phi u_i) = 0 \quad (9)$$

## 2.7 Experimentelle Ermittlung von Schiffswiderständen

Der klassische, und heute hauptsächlich zur Validierung von Simulationsergebnissen durchgeführte, Weg zur Ermittlung von Schiffswiderständen ist der Modellversuch. Dabei wird ein Modell des Schiffes in einem Maßstab  $\lambda$  gebaut und Versuche mit diesem durchgeführt. Um den Glattwasserwiderstand des Schiffes herauszufinden, wird der Schlepptank durchgeführt. Dabei wird der Rumpf des Modells durch einen Schlepptank gezogen und die Widerstandskraft gemessen. Der Vorgang ist nach Vorschriften der ITTC genormt, um Vergleichbarkeit zwischen Versuchen unterschiedlicher Institutionen herzustellen.

Mit entsprechenden Messvorrichtungen ist es auch möglich das Wellenbild und den Nachstrom zu erfassen. Die gewonnenen Ergebnisse müssen anschließend interpretiert und unter Beachtung von Modellgesetzen auf die Großausführung übertragen werden. Genaueres kann in dem Normenkatalog der ITTC [12] nachgelesen werden.

Das double-body flow Experiment dient ursprünglich der Ermittlung des Formfaktors  $k$ . Der Modellrumpf wird dabei auf Höhe der Wasserlinie in der xy-Ebene gespiegelt. Dieses Doppelrumpfmodell wird anschließend in einen Umlauftank gesetzt, um Widerstand und Nachstrom zu messen. Die Oberflächeneffekte werden somit bei diesem Experiment umgangen und es stellt sich eine stationäre Strömung ein.

### 3 Erstellung des numerischen Gitters

Der untersuchte Rumpf, das KRISO Containerschiff, wurde für Modelltests und CFD-Analysen vom "Korea Research Institute for Ships and Ocean Engineering" (heute MOERI) im Jahre 1997 entworfen, jedoch nie als Großausführung gebaut. Er hat eine Länge von 230m, eine Breite auf der Wasserlinie von 32,2m und einen Tiefgang von 10,8m. Der Rumpf ist auf eine Froude-Zahl von 0,26 ausgelegt, was bei der Großausführung einer Geschwindigkeit von 24 Knoten entspricht. Das Schiff kann 3600 TEU aufnehmen. Der Modellmaßstab  $\lambda$  in dieser Arbeit beträgt 31,6, die Modelllänge ist somit 7,2786m.

Die sechs für die Simulation verwendeten Gitter wurden mit OpenFOAM internen Programmen erstellt. Die Vorgehensweise war dabei immer gleich. Es wird ein blockstrukturiertes Gitter erstellt, in das anschließend die Geometriedatei des Schiffsrumpfes (im Dateiformat .stl) eingebunden wird. Im gleichen Schritt der Geometrie einbindung ist es möglich, sowohl das Gitter in Wandnähe oder andere festgelegte Gebiete weiter zu verfeinern, als auch "Layer" hinzuzufügen. "Layer" sind Zellschichten (siehe Abb. 2), die an eine Wand gelegt werden, um die Grenzschicht und deren Verhältnisse besser darstellen zu können.

Die Geometriedatei wurde von Dr.-Ing Gonzalo Tampier (Betreuer der Arbeit) zur Verfügung gestellt und ist in Abbildung 1 in verschiedenen Ansichten zu sehen. Der digitale Rumpf hat Modellgröße (7,2786m Länge), besitzt keine Anhänge und berücksichtigt bereits den dynamischen Trimm. Da das Modell symmetrisch ist, wird nur die Steuerbordhälfte in ein Gitter eingebunden.

Die Gittererstellung beinhaltet mehrere Schritte, die jeweils auf der Linux-Konsole eingegeben werden müssen. Um nicht jedes Mal die Befehlskette einzugeben, wurde ein aufrufbares Skript anhand eines existierenden Beispiels erstellt. Das Skript ist im Anhang angefügt (siehe Abschnitt D). Es nutzt den m4 Makros, um wiederholt vorkommende Größen (wie Längenangaben) in einer zentralen Datei ablegen zu können. Eine grundlegende Version dieses Skripts wurde von Dr.-Ing G. Tampier bereitgestellt.

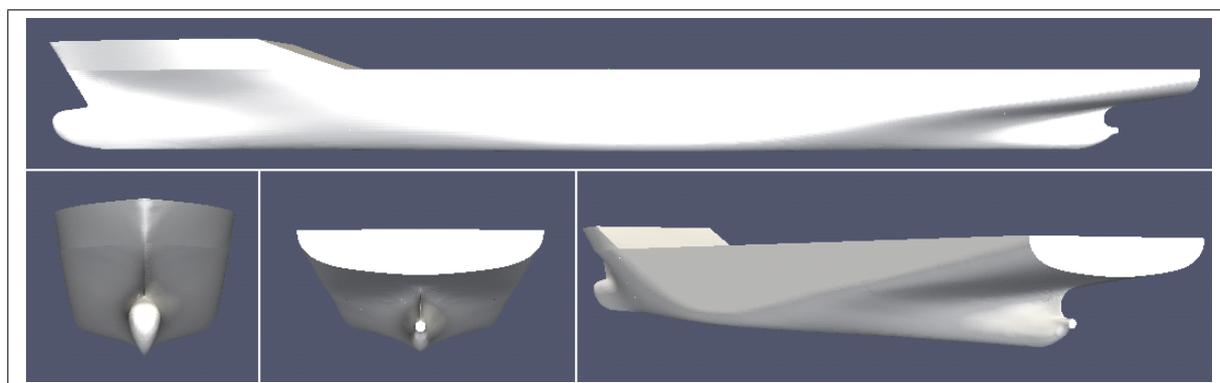


Abbildung 1 Ansicht der Modellgeometrie des KCS

Das blockstrukturierte Gitter wird mit dem OpenFOAM Programm `blockMesh` erstellt. Zunächst wird ein Gitter hexagonaler Zellen aus 17 Blöcken erstellt, welches eine Aussparung an der Stelle hat, wo der Rumpf eingesetzt wird. Jeder Block hat unterschiedliche Zellengrößen, je nach Funktion seiner jeweiligen Position (siehe Abb. 3). Der Vorlauf hat dabei nur Zellen mit gleichem Abstand, im Nachlauf werden die Zellen graduell länger. Auf diese Weise werden die Wellen zum Auslass hin "geglättet", um möglichst wenig Wellenreflexionen zu erhalten. Zum Gitterboden hin werden die Zellen auch größer, da dieses Gebiet von wenig Interesse ist und daher keine feine Auflösung benötigt. In der Breite (y-Richtung) wird der Zellenabstand auch langsam aber kontinuierlich größer. In einem darauffolgenden Schritt werden bestimmte Zonen weiter verfeinert. Hauptsächlich handelt es sich dabei um den Bereich der freien Oberfläche, damit die Wellen gut genug aufgelöst werden können. Die Gitterabmessungen sind in Tabelle 3 nachzulesen.

Parameter	Länge
Modelllänge ( $L_{PP}$ )	7,2786 m = L
Vorlauf	1,35 L
Nachlauf	2,85 L
Wassertiefe	8 m
Gittergesamtlänge (x)	38 m
Gitterbreite (y)	8 m
Gitterhöhe (z) (bei FO)	9 m

**Tabelle 3** allgemeine Gitterparameter

Die Rumpfgeometrie wird mit dem OpenFOAM Programm `snappyHexMesh` eingebunden. Dies ist ein skriptgesteuerter Generator von unstrukturierten Gittern. Dafür wird zuerst ein Gitterblock mit `blockMesh` erzeugt, der in die vorher freigelassene Lücke passt. Der Block wird dabei in allen Dimensionen 0,5% größer gemacht, damit sich die Gitter überlappen und anschließend zusammenfügen lassen.

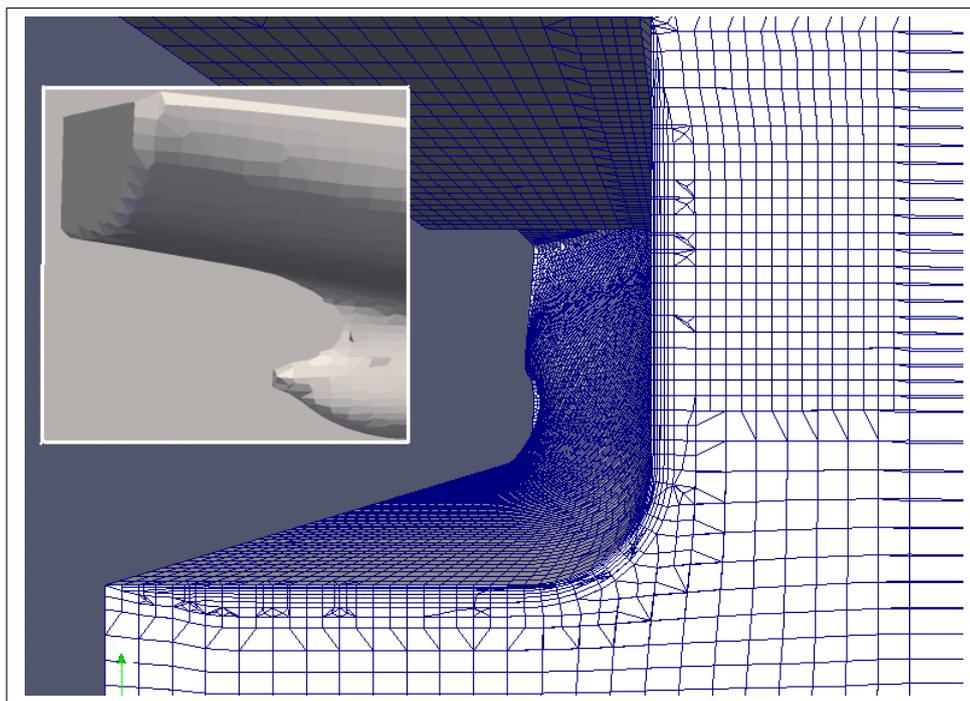
Nun bindet `snappyHexMesh` die Geometrie ein, indem es Zellen nahe der Geometrie zerteilt und die Gittergrenze(n) an die Oberfläche anpasst. Die genaue Funktionsweise von `snappyHexMesh` kann im OpenFOAM Handbuch nachgelesen werden [3].

Innerhalb vom Programmablauf von `snappyHexMesh` können auch Gitterregionen verfeinert und Layer aufgebaut werden. So wurde unter anderem der Bereich des Nachstroms bei den feinen Gittern angepasst, um eine genauere Auflösung der Strömungsphänomene zu erhalten. Die hinzugefügten Cell-Layer (Zellschichten) sind am Rumpf angelegt (siehe Abb. 2), um eine genauere Auflösung der Grenzschicht und somit der zugehörigen Geschwindigkeiten und Drücke zu ermöglichen. Die dreieckigen Zellen in Abbildung 2 sind Darstellungsfehler von `paraView` die entstehen, da das Programm nicht auf blockstrukturierte non-matching-grids ausgelegt ist.

Zum Schluss wird der Block mit der eingebundenen Geometriedatei in das vorher erstellte

Hauptgitter integriert. Gittergrenzen (boundaries) müssen in diesem Vorgang angepasst werden um ein einheitliches Gitter zu erhalten. Das Zusammenfügen der beiden Gitterteile kann teilweise auch nicht funktionieren, wenn die Zelleneinteilung in x-Richtung im Geometrieblock zu fein ist. Falls dies eintritt, muss eine günstige Zellenanzahl gefunden werden, bei der die Probleme nicht auftreten.

Dieser zweistufige Aufbau der Gittererzeugung ist notwendig, da `snappyHexMesh` vielfältige Fehler produzieren kann, wenn man es in einem komplizierteren Gitter ausführt. Ein weiteres Manko von `snappyHexMesh` ist, dass bei länglichen Geometrien wie einem Schiffsrumpf, die Zellen nicht sauber an die Oberfläche angepasst werden, sondern teilweise überstehen. Abhilfe verschafft das Skalieren (Verkleinern) der x-Dimension bis auf würfelähnliche Maße und anschließendes Rückskalieren nach dem `snappyHexMesh`-Durchlauf. Auch werden glatte Kanten während der Geometrie einbindung teilweise abgerundet, siehe kleines Bild in Abbildung 2. Detaillierte Informationen über Probleme und Möglichkeiten von `snappyHexMesh` in der numerischen Schiffshydrodynamik können im Paper von Tampier (2010) [14] nachgelesen werden.



**Abbildung 2** Layer im feinen Gitter mit FO: vertikaler Schnitt bei  $x=0$   
kl. Bild: Heckansicht nach `snappyHexMesh`

Nach dieser Gittergenerierungsroutine erhält man ein Gitter mit den jeweils gewählten Abmaßen. Die gleich bleibenden Maße der verwendeten Gitter können Tabelle 4 entnommen werden. Das Schiff ist so im Koordinatensystem positioniert, dass  $x=0$  sich mittschiffs befindet,  $y=0$  in der Symmetrieebene liegt und  $z=0$  auf Höhe der Ruhewasserlinie ist.

Insgesamt wurden jeweils drei Gitter mit und ohne freie Flüssigkeitsoberfläche erzeugt. Die Gitter wurden von 1 bis 3 durchnummeriert, wobei Gitter 1 das feinste und Gitter 3 das größte ist. Zwischen den Gittern wurde der Verfeinerungsgrad 1,5 angesetzt. Dies war aber nicht immer exakt realisierbar, wegen den oben angesprochenen Problemen mit `snappyHexMesh` und beim Zusammenfügen der Gitterteile.

Insgesamt ist die Qualität der Gitter sehr wichtig für die Genauigkeit der Ergebnisse. Sie fließt unter anderem mit einer Gitterunsicherheit in die Verifikation mit ein.

### 3.1 Gitter mit freier Oberfläche

Die drei Gitter mit freier Oberfläche wurden nach dem geschilderten Schema erstellt. Der Bereich der freien Oberfläche wurde dabei zusätzlich verfeinert um das Wellenbild genauer auflösen zu können, wie in Abbildung 3 zu sehen. Die genaue Anzahl an Zelltypen kann in Tabelle 4 abgelesen werden.

Zellen	Gitter 1	Gitter 2	Gitter 3
hexahedra	648622	177921	56594
prisms	703	180	86
tetWedges	16	3	11
polyhedra	51259	16003	6468
<b>Gesamtanzahl</b>	<b>700600</b>	<b>194107</b>	<b>63159</b>

Tabelle 4 Zelldaten der Gitter mit FO

Mit diesem Gitter werden die Simulationen mit freier Oberfläche durchgeführt. Wie sich die Grenze zwischen Wasser und Luft per VOF-Methode gesetzt wird, ist im nächsten Kapitel erklärt.

Bei den drei Gittern gibt es jeweils 7 Gittergrenzen (boundaries). Der Einlass, der Auslass, der äußere Rand (`yMax`) und die obere Grenze (`zMax`) sind generische Patches. Das heißt, sie enthalten keine Information über die Geometrie, sondern weisen das Lösungsprogramm nur auf eine Grenze hin. Die untere Grenze (`zMin`) und der Schiffskörper sind vom Typ "wall". Dadurch können bei Bedarf Wandfunktionen in OpenFOAM eingesetzt werden. Die Grenze im Mittschiffsbereich (`yMin`) ist eine Symmetrieebene.

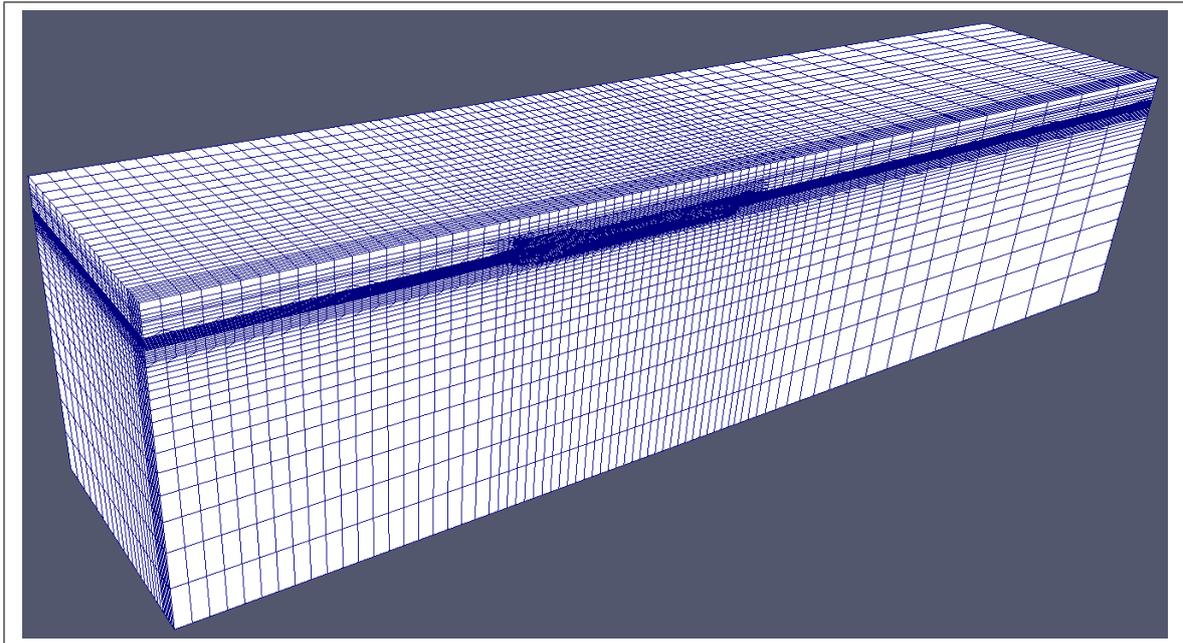


Abbildung 3 mittleres Gitter mit FO

### 3.2 Gitter ohne freie Oberfläche

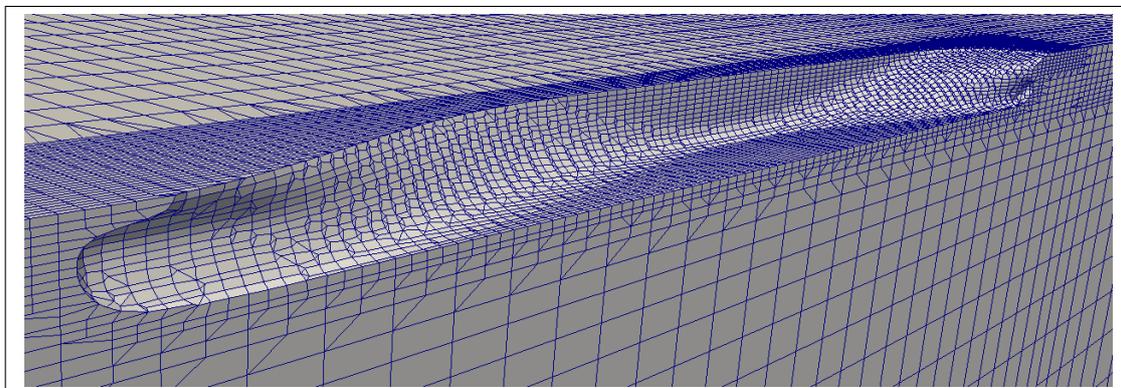
Für die Double-Body Simulation muss das Gitter auf Höhe der Wasserlinie enden. Dafür kann die Gitterhöhe so definiert werden, dass sie bei  $z = 0$  endet. Diese neue Grenze wird als Symmetrieebene deklariert.

Zellen	Gitter 1	Gitter 2	Gitter 3
hexahedra	272218	78715	27501
prisms	402	114	30
tetWedges	5	4	2
polyhedra	24131	9278	3527
<b>Gesamtanzahl</b>	<b>296756</b>	<b>88111</b>	<b>31060</b>

Tabelle 5 Zelldaten der Gitter ohne FO

Ein anderer möglicher Ansatz ist das Gitter bei  $z=0$  mit dem Befehl `subsetMesh` abzutrennen. Dafür wird ein Volumen per Befehl `cellSet` definiert, welches anschließend "ausgeschnitten" wird. Den dabei neu entstandenen Randzellen muss nun eine Grenzeigenschaft per Befehl `createPatch` zugewiesen werden, in diesem Fall eine Symmetrieebene. Das Ganze muss vor dem Starten von `snappyHexMesh` geschehen. Da bei dieser Methode teilweise

Probleme, wie eine Stufe zwischen Innenbox und Außengitter, entstanden sind, wurde die erste Lösung bevorzugt genutzt.



**Abbildung 4** mittleres Gitter ohne FO

Am Ende wurden auf diese Weise wieder drei Gitter erstellt, wobei Gitter 1 das feinste ist. Die genauen Zellanzahlen der Gitter sind in Tabelle 5 zu sehen. Beim Gitter 1 wurde zusätzlich der Nachstrombereich feiner aufgelöst, um die Simulationsergebnisse besser mit dem Experiment vergleichen zu können. Wie der Schiffsrumpf im Gitter ohne FO eingefügt ist, kann man in Abbildung 4 sehen. Die Zellen in Rumpfnähe wurden wie beim Gitter mit FO verfeinert, sowie Layer hinzugefügt. Die Region, wo vorher die freie Oberfläche war aber nicht, da hier keine Wellen aufgelöst werden müssen. Da dieser stark verfeinerte Bereich wegfällt und natürlich alle Zellen über  $z=0$ , ist die Gesamtzellenanzahl um einiges kleiner als bei dem jeweiligen Gitter mit freier Oberfläche.

## 4 CFD mit OpenFOAM 1.6

OpenFOAM ist ein kostenloses Open Source CFD-Softwarepaket. Der Quellcode (in der Programmiersprache C++) ist somit frei zugänglich und wurde hauptsächlich von der Firma OpenCFD Ltd geschrieben. OpenFOAM enthält eine Menge Programme mit Lösungsalgorithmen, sowie Programme für Vor- und Nachbearbeitung der Simulation. Diese sind modular aufgebaut und lassen sich somit einfach ergänzen und erweitern. Lösungsprogramme sind vorhanden für kompressible und inkompressible Strömungen, Mehrphasenströmung, DNS, Verbrennungsvorgänge, Wärmeleitung, Spannungsanalyse bei Festkörpern und weitere Anwendungsfälle. [10]

OpenFOAM wird auf einem Linux-Betriebssystem installiert und per Kommandozeile bedient. Eine grafische Oberfläche als solche gibt es nicht. Alle Informationen sowie Einstellungen werden über Einträge in Textdateien gesteuert. Um Simulationsergebnisse zu visualisieren, kann das "post-processing" Tool `paraView` eingesetzt werden. Dies ist ein sehr vielfältiges Datenanalyse- und Visualisierungsprogramm, mit dem alle Bilder der Gitterstruktur und der Ergebnisse dieser Arbeit erstellt wurden.

Für die durchgeführten Simulationen wurden zwei der verschiedenen Lösungsprogramme in OpenFOAM genutzt, für die Double-Body Simulation das Lösungsprogramm für inkompressible Einphasenströmungen `simpleFoam` und für die Simulation mit freier Oberfläche das Programm für inkompressible Mehrphasenströmungen (mit der VOF-Methode) `interFoam`.

### 4.1 Dateistruktur von OpenFOAM 1.6

Um in OpenFOAM Simulationen starten zu können ist eine bestimmte, immer gleich bleibende Ordnerstruktur inklusive enthaltenen Dateien notwendig. Diese Ordner tragen die Namen *0*, *constant* und *system*.

#### **Ordner 0:**

In diesem Ordner befindet sich die Definition der Anfangs- und Randbedingungen der Simulation. Diese umfassen alle physikalischen und Modellgrößen wie  $k$  (kinetische Energie der Turbulenz),  $\nu_t$  (Wirbelviskosität),  $\omega$  (turbulente Frequenz),  $U$  (Geschwindigkeit),  $p$  (Druck) und andere eventuell notwendige Größen. Für jede Gittergrenze muss eine Randbedingung angegeben werden sowie für das interne Feld eine Anfangsgröße. Die Randbedingungen der Simulationen befinden sich im Anhang in Abschnitt B, siehe Tabelle 14 und 15.

#### **Ordner constant:**

Hier befinden sich alle Daten, die über den Simulationsprozess konstant bleiben. Das Gitter liegt in dem Unterordner *polyMesh*, darin liegt auch die Definitionsdatei der Gittergrenzen

(*boundary*), in der die Namen der Grenzen geändert und die Art der Grenze definiert werden kann. Weiterhin enthält der *constant* Ordner die Transporteigenschaften der Simulation wie RASModell, Fluideigenschaften und verwendetes Turbulenzmodell.

### Ordner *system*:

Dies ist der Ordner, in dem alle Daten zur Steuerung der Simulation liegen. Die wichtigste Datei ist dabei *controlDict*, hier werden unter anderem das Lösungsprogramm, die Größe und Anzahl der Zeitschritte und die Abstände der Ausgabe von Rechenergebnissen definiert. Im Ordner *system* werden auch die Lösungsschemen und Lösungsalgorithmen für alle auftretenden Größen festgelegt. Zusätzlich können hier verschiedene "dict"-Dateien liegen, die für pre- und post-processing Tools notwendig sind. Ein Beispiel hierfür ist die *mapFieldsDict* Datei, die für das Übertragen von früheren Simulationsergebnissen (beispielsweise eines gröberen Gitters) in die Anfangsbedingungen der aktuellen Simulation nötig ist.

Um die Rechenzeit der einzelnen Simulationen zu verringern, wurde parallel gerechnet. Dafür ist die "dict"-Datei *decomposeParDict* notwendig. Hier wird festgelegt, auf wie viele Prozessoren der Rechenaufwand verteilt wird. Anschließend kann mit den Befehlen *decomposePar* und *reconstructPar* das Gitter in Rechengebiete unterteilt und wieder zusammengesetzt werden. Für die Simulationen in dieser Arbeit wurde jeweils auf 8 Kernen gerechnet. Alle Rechnungen wurden auf dem Cluster des Fachgebiets Dynamik Maritimer Systeme mit dem Namen "dmscluster01" durchgeführt. Der Cluster-Rechner verfügt insgesamt über sieben zusammengeschlossene Computer mit jeweils acht Prozessoren, wobei maximal auf 48 Prozessoren gleichzeitig gerechnet werden kann.

## 4.2 Einstellungen: Simulation ohne freie Oberfläche

Für die Double-Body Simulation wurde das Lösungsprogramm *simpleFoam* genutzt, da es sich um eine einphasige, inkompressible, turbulente Strömung handelt. Das verwendete Turbulenzmodell ist *komegaSST*. Die Rand-/Anfangsbedingungen der Simulation (Dateien im Ordner *0*) sind in Tabelle 14 zu sehen.

Die Anströmgeschwindigkeit  $U = 2,1966 \frac{m}{s}$  wurde über die vorgegebene Froude-Zahl bestimmt (siehe Anhang Abschnitt A). Der Anfangswert für  $k$  wurde über die von Ferziger/Peric [5] empfohlene Näherungsgleichung bestimmt. Die Turbulenzintensität  $I_t$  wird dabei mit 1% angenommen.

$$k \approx \frac{3}{2} I_t^2 U^2 = \frac{3}{2} \cdot (0,01)^2 \cdot \left(2,1966 \frac{m}{s}\right)^2 = 7,2 \cdot 10^{-4} \frac{m^2}{s^2} \quad (10)$$

Die Wirbelviskosität  $\nu_t$  wird mit dem Zehnfachen der Molekularviskosität angenommen [5]. Damit kann ein Startwert für  $\omega$  bestimmt werden.

$$\omega = \frac{k}{\nu_t} = \frac{7,2 \cdot 10^{-4} \frac{m^2}{s^2}}{1 \cdot 10^{-6} \frac{m^2}{s}} = 720 \frac{1}{s} \quad (11)$$

Die Wirbelviskosität  $\nu_t$  muss in OpenFOAM nicht definiert werden, da sie aus omega und  $k$  errechnet werden kann (umgestellte Gl. 11).

Das verwendete komegaSST Turbulenzmodell muss im *constant* Ordner vermerkt sein. Das verwendete Gitter wurde bereits im vorangegangenen Abschnitt beschrieben.

Außerdem muss eine Schrittgröße für die Zeit festgelegt werden. Dies geschieht mit Hilfe der Courant-Zahl  $Co$ . Sie stellt ein Verhältnis zwischen Geschwindigkeit und Zellenlänge dar und ist wie folgt definiert.

$$Co = \frac{\delta t |\vec{U}|}{\delta x} \quad (12)$$

$\delta t$  in der Gleichung (12) ist der Zeitschritt,  $U$  ist die Geschwindigkeit und  $\delta x$  die Zellenlänge. Die Courant-Zahl sollte auf jeden Fall unter 1 und möglichst unter 0,5 in jeder Zelle bleiben. Der Zeitschritt wurde mit  $\delta t = 0,001$  Sekunden gewählt, da man mit der Annahme  $U_{max} = 2,3 \frac{m}{s}$  bei Zellenlängen bis 5mm auf jeden Fall unter  $Co = 0,5$  bleibt. Dabei ist 5mm Zelllänge sehr klein und wird in den realisierten Gittern auf keinen Fall erreicht.

Die festgelegten Lösungsschemen für die einzelnen Rechenoperationen sind in Tabelle 6 aufgeführt. Diese Informationen sind in der Datei *fvSchemes* im Ordner *system* abgelegt. Die Lösungsschemen sind entscheidend für ein gutes Ergebnis. Sie wurden größtenteils aus Beispielfällen in OpenFOAM übernommen und mit Empfehlungen des Betreuers Dr.-Ing Gonzalo Tampier abgeändert bzw. ergänzt. Einige Lösungsschemen können in bestimmten Fällen zur Divergenz der Lösung führen und man muss die Simulation mit simpleren, unanfälligeren Schemen initialisieren. Dies wird in Abschnitt 4.5 nochmal aufgegriffen.

Ausdruck	Lösungsschema
alle Gradienten	Gauss linear
div(phi,U)	Gauss vanLeerV
div((nuEff*dev(grad(U).T())))	Gauss linear
andere Divergenzen	Gauss upwind
alle Laplaceoperatoren	Gauss linear corrected
alle Interpolationsschemen	linear

**Tabelle 6 Lösungsschemen der Simulation ohne FO**

Die Lösungsalgorithmen wurden alle aus OpenFOAM Beispielfällen übernommen. Da sich bei dieser Auswahl keine Probleme ergaben, wurde sie beibehalten. Der einzige Parameter, an dem Änderungen vorgenommen wurden, ist der Unterrelaxation-Wert. Unterrelaxation wird bei iterativen Lösungsmethoden eingesetzt, um die Änderung einer Variablen zwischen den Iterationsschritten zu dämpfen. Dadurch kann bei komplexen Problemen das Divergieren der Lösung vermieden werden. Eine Übersicht der verwendeten Lösungsalgorithmen mit Unterrelaxationsfaktoren (stammend aus der Datei *fvSolution*) sieht man in Tabelle 7.

Größe	Lösungsalg.	smoother	Preconditioner	Unterrel.-Faktor
p	GAMG	GaussSeidel	-	0,5
U	smoothSolver	GaussSeidel	-	0,6
k	smoothSolver	GaussSeidel	-	0,6
omega	smoothSolver	GaussSeidel	-	0,6
Druck-Geschw.	SIMPLE	-	-	-

Tabelle 7 Lösungsalgorithmen der Simulation ohne FO

### 4.3 Einstellungen: Simulation mit freier Oberfläche

Bei der Simulation mit freier Oberfläche wurde das Lösungsprogramm `interFoam` genutzt, das zweiphasige, inkompressible, turbulente Strömungen berechnen kann und die Phasen per VOF-Methode trennt. Die Anfangsbedingungen (Ordner  $\theta$ ) sind ähnlich denen der Simulation ohne freie Oberfläche. Eine Übersicht der Randbedingungen ist im Anhang in Tabelle 15 zu sehen. Hinzu kommt hierbei die Definition der Anfangsposition der Phasen. Dafür wird in OpenFOAM die Variable `alpha1` ( $\alpha_1$ ) eingeführt. Wenn der Wert von  $\alpha_1$  gleich 0 in einer Zelle ist, beinhaltet diese nur Luft, falls der Wert 1 ist, beinhaltet sie nur Wasser. Bei  $\alpha_1$  gleich 0,5 befindet sich die freie Oberfläche. Mit dem OpenFOAM Tool `setFields` wird das Volumen, nach vorab (in der `setFieldsDict` Datei) definierten Anteilen, in die zwei Phasen eingeteilt.

Das Turbulenzmodell `komegaSST` wird mit den gleichen Einstellungen wie bei der Simulation ohne FO verwendet. Der Zeitschritt wurde auf  $\delta t = 0,001$  festgelegt, wobei nach 10 Sekunden Simulationslaufzeit eine OpenFOAM Option genutzt wurde, die den Zeitschritt an eine maximal vorgegebene Courant-Zahl anpasst. Dadurch kann, nach anfänglicher Stabilisierung der Lösung, ein größerer Zeitabschnitt mit weniger Rechenschritten als bei fester Courant-Zahl gelöst werden. Warum dies nötig war, wird in Abschnitt 4.5 besprochen.

Ausdruck	Lösungsschema
alle Zeitableitungen	Euler
alle Gradienten	Gauss linear
<code>div(rho*phi,U)</code>	Gauss vanLeerV
<code>div(phi,alpha)</code>	Gauss vanLeer
<code>div(phirb,alpha)</code>	Gauss interfaceCompression
andere Divergenzen	Gauss linear
alle Laplaceoperatoren	Gauss linear corrected
alle Interpolationsschemen	linear

Tabelle 8 Lösungsschemen der Simulation mit FO

Die Lösungsschemen (in der Datei *fvSchemes*) sind in Tabelle 8 aufgeführt. Diese sind grundlegend gleich den Einstellungen ohne FO, nur dass die Zeitableitungen und Rechenvorgänge der Transportgleichung von  $\alpha_1$  hinzukommen.

Die Lösungsalgorithmen sind in Tabelle 9 aufgeführt, sie gleichen denen der Simulation ohne FO. Die einzige Änderung ist, dass für die Druck-Geschwindigkeit Kopplung PISO verwendet wurde, da *interFoam* dies erfordert. Der PISO Algorithmus in OpenFOAM beinhaltet die Korrektur des  $\alpha_1$ -Feldes. Der Wert des Parameters *cAlpha* in der Datei *fvSolution* gibt dabei die Kompression der Oberfläche an. Er wurde auf 0 gesetzt (keine Kompression), da andere Einstellungen zeitweise zu Divergenz führten.

Die Größen "pcorr" und "pfinal" kommen hinzu, da *interFoam* sie zur Lösung der zweiphasigen Strömung benötigt. Die Lösungsalgorithmen für diese wurden auch aus Beispielfällen in OpenFOAM entnommen.

Größe	Lösungsalg.	smoother	Preconditioner	Unterrel.-Faktor
p	GAMG	GaussSeidel	-	0,5
pcorr	PCG	-	DIC	0,5
pfinal	PCG	-	DIC	0,5
U	smoothSolver	GaussSeidel	-	0,5
k	smoothSolver	GaussSeidel	-	0,5
omega	smoothSolver	GaussSeidel	-	0,5
Druck-Geschw.	PISO	-	-	-

Tabelle 9 Lösungsalgorithmen der Simulation mit FO

## 4.4 Berechnen der Druck- und Reibungskräfte

Die Kräfte wurden mit der OpenFOAM internen Funktion *forces* ermittelt. Diese integriert die Spannungen über eine Fläche, um die Druck- und Reibungskräfte zu erhalten. Die Funktion kann für jeden Zeitschritt die Kräfte am Rumpf in eine Datei schreiben. Die gespeicherten Kräfte sind zerlegt in kartesische Koordinaten. Interessant für den Glattwasserwiderstand ist nur die x-Komponente. Momente in Bezug auf einen festgelegten Punkt kann *forces* bei Bedarf auch bestimmen.

Bei zweiphasiger Strömung wird die Funktion *forcesInter* verwendet. Dies ist eine Abwandlung von *forces*, die nur die Kräfte berechnet, die das Wasser verursacht. Dafür wird der Kraftwert einer Zelle mit dem Phasenwert  $\alpha_1$  multipliziert ( $\alpha_1 = 1$  entspricht Wasser).

Die *forces/forcesInter* Funktion muss in der Datei *controlDict* aufgerufen werden. Dafür werden einige Zeilen hinzugefügt, die im Anhang in Abschnitt D aufgeführt sind.

## 4.5 Aufgetretene Probleme bei der Simulation

Es trat ab und zu die Situation auf, dass die Lösung der Geschwindigkeit und auch die anderer Variablen, gegen unendlich divergierte. Vor allem in den ersten Sekunden der Simulationen mit freier Oberfläche. Dies ist teilweise durch das mit `snappyHexMesh` generierte Gitter bedingt. Der automatische Gittererstellungsmechanismus gestaltet z.B. die Zellen am Spiegelheck nicht sehr sauber, wodurch Zellen auch schräg angeströmt werden. Auch die zu Simulationsbeginn entstehende Bugwelle kann Divergenz der Lösung zur Folge haben.

Die Lösung dafür war ein Start der Simulation mit dem sehr stabilen Lösungsschema "Gauss upwind" für den Ausdruck  $\text{div}(\rho \cdot \phi, U)$ . Nach einigen Sekunden (z.B. 10s, ist aber auch früher möglich) kann die Simulation unterbrochen und das Lösungsschema "Gauss vanLeerV" eingesetzt werden.

Eine weitere, sehr unangenehme Problematik von OpenFOAM ist eine reflektierende Welle in der Simulation mit freier Oberfläche. Wenn die Simulation startet, bildet sich kurzzeitig eine vergleichsweise große initiale Bugwelle. Diese Welle wird anschließend von Auslass und Einlass reflektiert. Sie verkleinert sich nur sehr langsam und bleibt somit lange erhalten. Dadurch verändert sich das Wellenbild leicht, vor allem aber schwankt der Druckwiderstand des Rumpfes periodisch.

Um diesen Effekt möglichst klein zu halten, ist es üblich sehr grobe Zellen am Auslass zu verwenden. Diese glätten die Wellen am Auslass und minimieren/eliminieren ihre Reflexion. Diese groben Zellen sind in allen verwendeten Gittern implementiert, liefern aber nicht das gewünschte Ergebnis. Daher wurde die Simulation des mittleren Gitters probeweise bis 200 Sekunden durchgeführt, um zu sehen was mit der Welle passiert. Diese verkleinert sich langsam und die Schwankung der Druckkraft strebt gegen einen Mittelwert. Insgesamt hat die Welle aber auch nach 200s einen nennenswerten Einfluss auf das Gesamtergebnis. Auch eine Initialisierung der Simulation mit dem Strömungszustand des groben Gitters (per Funktion `mapFields`) enthält diese reflektierende Welle im annähernd gleichen Maß. Diese Problematik in OpenFOAM muss auf jeden Fall weiter untersucht und behoben oder umgangen werden.

## 5 Simulationsergebnisse

Alle Ergebnisse in Wellen- und Nachstrombildern wurden mit dem OpenFOAM internen Programm `paraView` erstellt. Um den direkten und unverfälschten Vergleich mit den experimentellen Daten der KCS Schleppversuche (siehe Website [1]) zu ermöglichen, wurden die online verfügbaren Felddaten der KCS-Website in ein für `paraView` lesbares Format umgewandelt (VTK-Dateien).

Die auszuwertenden Ergebnisse der Rechnung ohne FO (freie Flüssigkeitsoberfläche) sind der Formfaktor  $k$ , der viskose Widerstandsbeiwert  $C_V$ , das Nachstromfeld bei  $x/L=0,4911$  und der axiale Geschwindigkeitsschnitt bei  $x/L=0,4911$  und  $z/L=-0,03$ . Anzumerken ist, dass der Nachstrom nur in dem Gitter ohne FO ausgewertet wird, da diese Rechnung komplett stationär ist (gelöst mit `simpleFoam`) und nicht, wie beim Gitter mit FO, die reflektierende Welle enthält, die den Nachstrom verfälschen kann.

Bei der Rechnung mit FO sind der Gesamtwiderstandsbeiwert  $C_T$ , der Druckwiderstandsbeiwert  $C_P$ , das Wellenbild und der Wellenlängenschnitt bei  $y/L=0,1509$  auszuwerten.

### 5.1 Rechnungen ohne freie Flüssigkeitsoberfläche

Der Formfaktor  $k$  ist die prozentuale Erhöhung des Reibungswiderstandes durch zusätzlichen viskosen Druckwiderstand, verursacht durch die dreidimensionale Form des Schiffes. Er kann bei diesen Rechnungen wie folgt bestimmt werden:

$$k = \frac{C_{PV}}{C_F} \quad (13)$$

Die dafür benötigten Werte können direkt aus der von `forcesInter` erstellten Datei ausgelesen werden, da der Druckwiderstand ohne FO nur durch viskose Einflüsse zustande kommen kann.

Der viskose Widerstandsbeiwert  $C_V$  ist die Addition des Reibungsbeiwertes  $C_F$  und des viskosen Druckbeiwertes  $C_{PV}$  und wird hier wie folgt bestimmt:

$$C_V = \frac{R_F + R_{PV}}{\frac{\rho}{2} \cdot v^2 \cdot S_0} \quad (14)$$

Variable	Gitter 1	Gitter 2	Gitter 3
$k$ in %	22,11	30,78	31,57
$C_V$	$3,29 \cdot 10^{-3}$	$3,44 \cdot 10^{-3}$	$3,425 \cdot 10^{-3}$

Tabelle 10 Ergebnisse von  $k$  und  $C_V$  für Gitter ohne FO

In Tabelle 10 sind die Ergebnisse der ganzheitlichen Variablen dargestellt. Dabei ist Gitter 1 das feinste und Gitter 3 das gröbste. Die dargestellten Ergebnisse sind jeweils Mittelungen über eine Sekunde. Dabei treten bei den Daten nur sehr kleine Schwankungen im Bereich von 0,05%  $S$  (Lösungswert) auf.

Die Nachstromfelder in den Abbildungen 5 und 6 wurden jeweils gegenübergestellt, um die Verbesserung vom groben zum feinen Gitter zu verdeutlichen. Damit eine bessere Vergleichbarkeit zwischen feinem Gitter und Experiment erreicht wird, wurde der Heckbereich in Gitter 1 etwas genauer verfeinert. Man sieht, wie sich der Heckwirbel im feinen Gitter vergrößert und an die experimentellen Daten annähert.

Der axiale Geschwindigkeitsschnitt in Abbildung 7 stellt die drei Gitter nebeneinander, jeweils im Vergleich zu den experimentellen Werten. Der per CFD ermittelte Geschwindigkeitsverlauf ist in Linienform, der experimentelle Verlauf zum Vergleich mit Punktwerten als Symbol dargestellt. Dabei sind die Größen  $V1$ ,  $V2$ ,  $V3$  die mit der Schiffsgeschwindigkeit ( $U$ ) normierten, axialen Geschwindigkeiten.

$$V1 = \frac{u}{U} \quad ; \quad V2 = \frac{v}{U} \quad ; \quad V3 = \frac{w}{U}$$

Man sieht, wie sich die Geschwindigkeit  $V1$  mit feiner werdendem Gitter an den experimentell ermittelten Geschwindigkeitsverlauf annähert. Diese Geschwindigkeit in x-Richtung ist auch die entscheidende für eine Propellerauslegung.

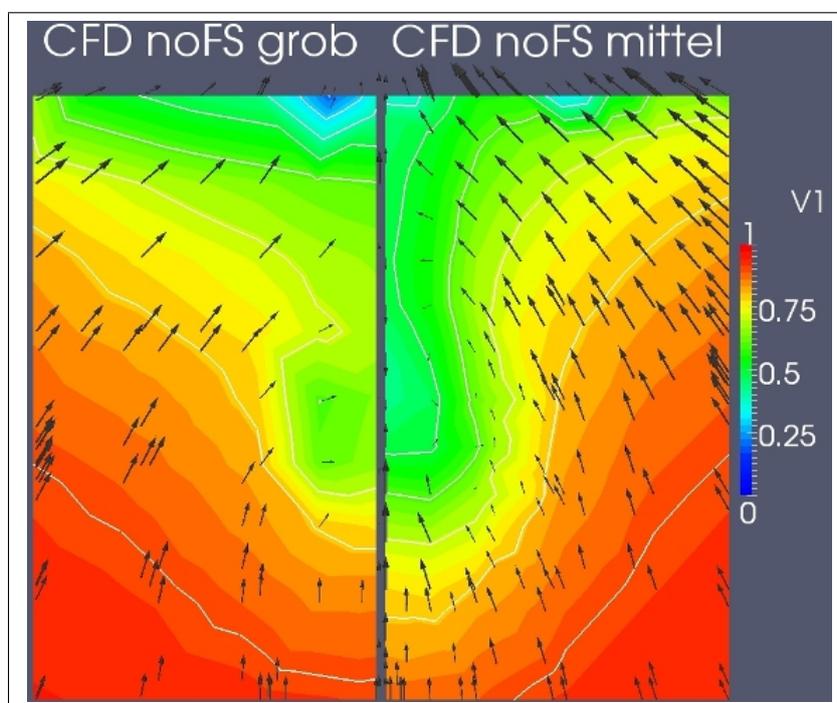


Abbildung 5 Nachstromfeld ohne FO, links Gitter 3, rechts Gitter 2

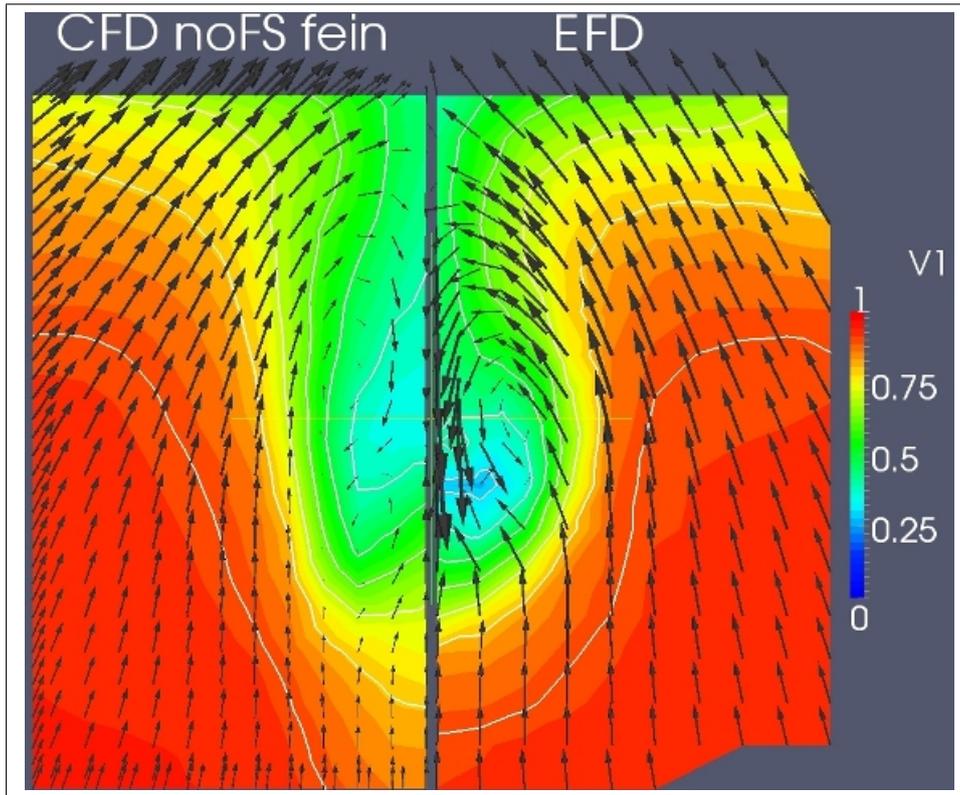


Abbildung 6 Nachstromfeld ohne FO, links Gitter 1, rechts EFD (Experiment)

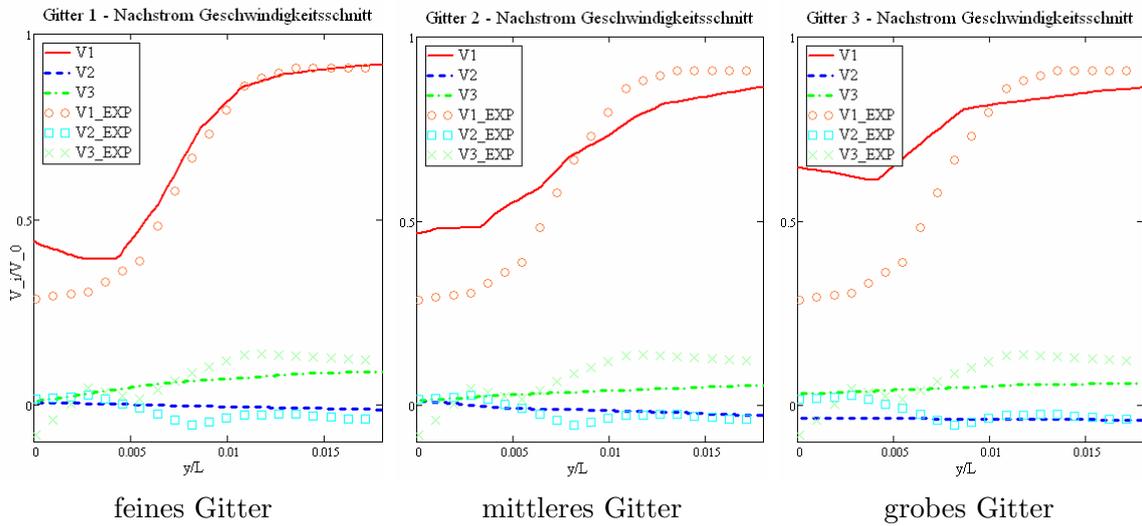


Abbildung 7 axiale Geschwindigkeitsschnitt bei  $x/L=0,4911$  und  $z/L=-0,03$

## 5.2 Rechnungen mit freier Flüssigkeitsoberfläche

Alle vom Druck beeinflussten Größen bei der Rechnung mit FO schwanken erheblich, da sich die vorher beschriebene reflektierende Welle sehr lange aufrecht erhält. Daher wurden die Ergebnisse jeweils über zwei Schwingungsperioden gemittelt.

Der Gesamtwiderstandsbeiwert  $C_T$  ist die Addition der Druckkräfte ( $R_P$ ) und Reibungskräfte ( $R_F$ ), normiert mit dem Staudruck mal benetzter Rumpffläche (Gl. 15). Die Kräfte können dabei direkt aus der von `forcesInter` erstellten Datei abgelesen werden. Durch die reflektierende Welle bleiben vergleichsweise große Schwankungen (10% von  $S_{C_T}$ ) sogar nach 50000 Iterationen noch erhalten (siehe Abb. 18). Dieses Problem tritt so stark bei keinem anderen CFD-Programm auf und bedarf weiterer Untersuchung, bzw. Weiterentwicklung des Codes. Diese großen Schwankungen bei den Kräften betreffen ausschließlich die Druckkraft. Sie bauen sich langsam ab, jedoch würde es schätzungsweise eine Million Iterationen benötigen, bis diese ganz abgebaut sind. So viele Iterationen und entsprechend lange Rechenzeiten sind nicht vertretbar und somit wurde auf die Mittelung über die erwähnten zwei Schwingungsperioden zurückgegriffen.

$$C_T = \frac{R_P + R_F}{\frac{\rho}{2} \cdot U^2 \cdot S_0} \quad (15)$$

Die Ergebnisse der globalen Variablen sind in Tabelle 11 zu sehen. Der Restwiderstandsbeiwert wurde mit Gleichung (16) ermittelt. Wie bei allen ausgewerteten Variablen, wird ein Ergebnis für jedes Gitter bestimmt.

$$C_R = C_T - (1 + k) C_F \quad (16)$$

Variable	Gitter 1	Gitter 2	Gitter 3
$C_T$	$4,187 \cdot 10^{-3}$	$4,473 \cdot 10^{-3}$	$5,45 \cdot 10^{-3}$
$C_R$	$0,535 \cdot 10^{-3}$	$0,498 \cdot 10^{-3}$	$1,519 \cdot 10^{-3}$

**Tabelle 11 Ergebnisse von  $C_T$  und  $C_R$  für Gitter mit FO**

Die Wellenbilder in Abbildung 8 und 9 (wie auch die Nachstromfelder, siehe Abb. 5 und 6) zeigen jeweils die Gegenüberstellung zweier Gitterergebnisse. Dabei ist der Maßstab "waveH" die Wellenhöhe in Metern. Eine Eliminierung der Dimension mit der Schiffslänge ist hier nicht notwendig, da das Wellenfeld nicht validiert wird (sondern nur der Wellenlängsschnitt, siehe Abb. 10). Die Abbildungen der Wellenfelder dienen lediglich der Visualisierung, um Änderungen der Wellenausprägung mit zunehmender Zellenanzahl und die Annäherung an das Experiment deutlich zu machen.

Es ist zu sehen, wie sich das berechnete Wellenbild mit feinerem Gitter immer weiter an

das experimentelle Ergebnis annähert und beim feinen Gitter optisch fast identisch ist. Die Isolinien wurden dabei nach optischen Kriterien so gewählt, dass man Ähnlichkeiten bzw. Abweichungen möglichst einfach erkennen kann. Die Abstufung der Isolinien ist folgende: 18 Linien zwischen den Werten  $-0,04\text{m}$  bis  $0,08\text{m}$  in gleichmäßigen Abständen verteilt.

In Abbildung 10 sind die Wellenlängsschnitte der drei Gitter und die experimentellen Ergebnisse in einem Diagramm zusammengefasst. Der Längenmaßstab, sowie die Wellenhöhe sind dabei mit der Länge des Modells normiert. Interessant am Diagramm ist, dass im Bugbereich der Wellenverlauf des Gitters 2 überraschend genau mit dem Experiment übereinstimmt. Wenn man aber das gesamte Wellenfeld des Gitters 2 (Abb. 8) mit dem Experiment (Abb. 9) vergleicht, sieht man, dass sich die Wellen nur im Bugbereich ähneln und an allen anderen Stellen im Vergleich zu Gitter 1 ungenauer sind.

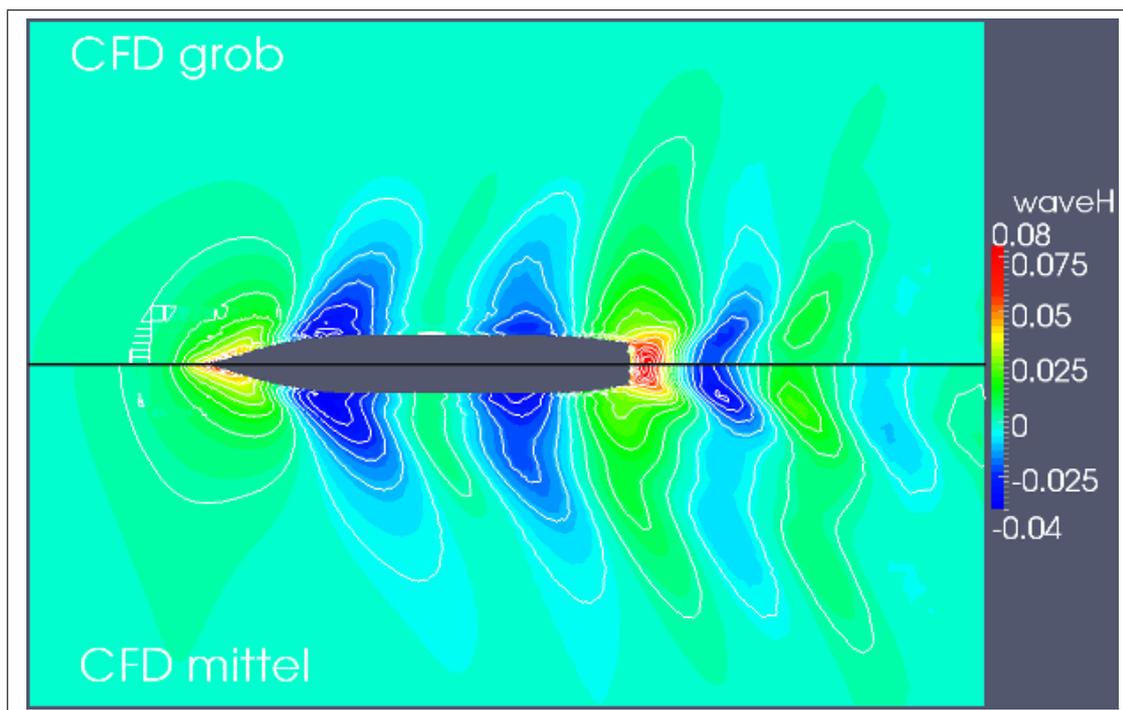


Abbildung 8 Wellenfeld, oben Gitter 3, unten Gitter 2; waveH in m

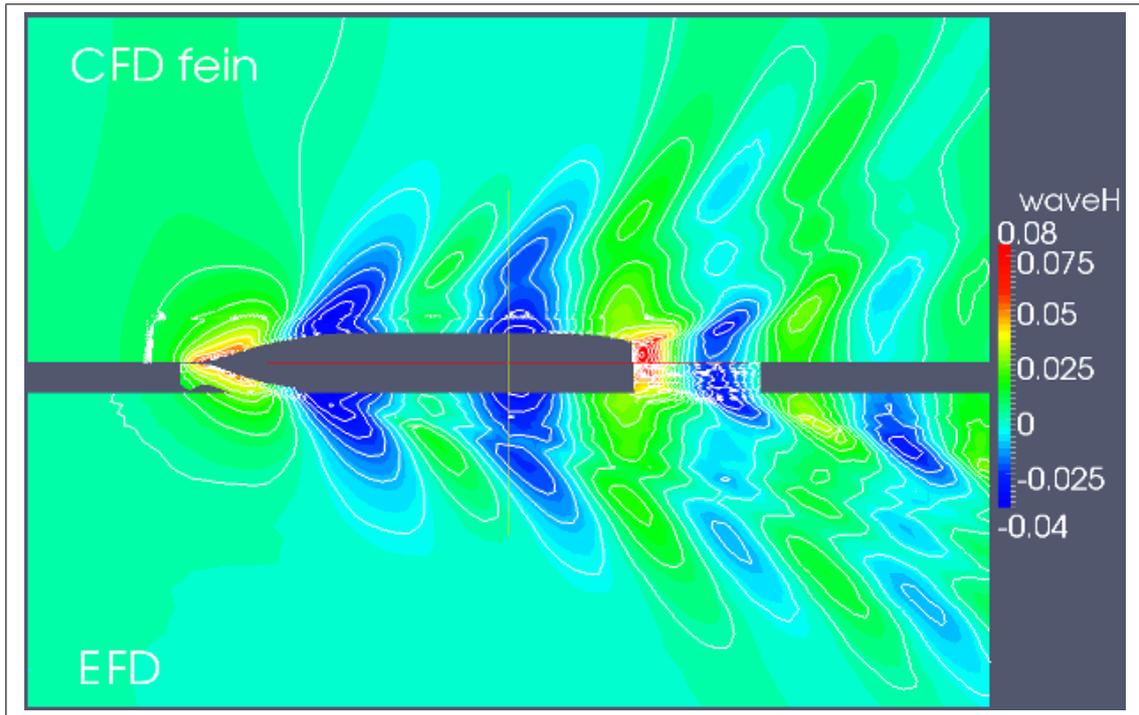


Abbildung 9 Wellenfeld, oben Gitter 1, unten EFD; waveH in m

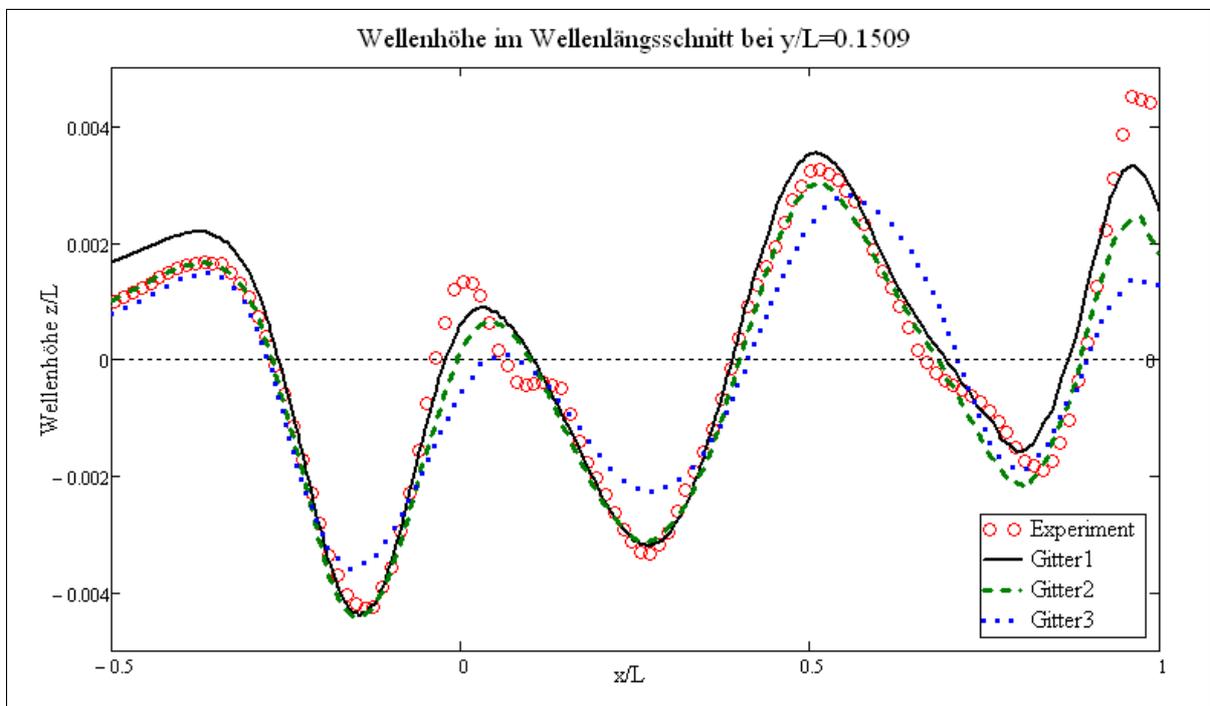


Abbildung 10 Wellenhöhe  $z/L$  im Längsschnitt bei  $y/L=0,1509$

## 6 Verifikation und Validierung

Vorab eine kurze Definition der beiden Begriffe:

**Verifikation** ist der Prozess der Beurteilung der numerischen Ungewissheit und, wenn die Gegebenheiten es erlauben, die Abschätzung der Größe und des Vorzeichens vom numerischen Fehler, sowie der Ungewissheit in der Fehlerabschätzung. [13]

**Validierung** ist der Prozess der Beurteilung der Unsicherheit des Modells mittels Nutzung von experimentellen Eckdaten und, wenn die Gegebenheiten es erlauben, die Abschätzung des numerischen Fehlers an sich. [13]

Das Vorgehen innerhalb der Verifikation und Validierung wird nach den Empfehlungen der ITTC (Uncertainty Analysis in CFD [2]) durchgeführt. Dabei werden Fehlergrößen und Unsicherheitsfaktoren ermittelt, um eine Aussage über die Qualität der Berechnungen treffen zu können. Der Simulationsfehler  $\delta_S$  ist definiert als Differenz zwischen Simulationsergebnis ( $S$ ) und Realität ( $T$  engl. truth). Gleichzeitig ist er die Addition des Modellfehlers ( $\delta_{SM}$ ) und des numerischen Fehlers ( $\delta_{SN}$ ). Leider ist  $\delta_S$  nie exakt bestimmbar, da das reale Ergebnis nicht zur Verfügung steht, sondern ein experimentell ermitteltes Ergebnis mit einer bestimmten Ungenauigkeit.

$$\delta_S = S - T = \delta_{SM} + \delta_{SN} \quad (17)$$

In einigen Fällen können die Größe und das Vorzeichen des numerischen Fehlers ( $\delta_{SN} = \delta_{SN}^* + \varepsilon_{SN}$ ) abgeschätzt werden. Dabei ist  $\delta_{SN}^*$  die Abschätzung des Fehlers und  $\varepsilon_{SN}$  der Fehler darin. Wenn dies zutrifft, kann ein korrigierter numerischer Bezugswert  $S_C$  verwendet werden:

$$S_C = S - \delta_{SN}^* \quad (18)$$

Anschließend wird die korrigierte numerische Unsicherheit der Simulation ( $U_{SCN}$ ) für weitere Vergleiche verwendet.

Für den nicht korrigierten Fall kann der numerische Fehler  $\delta_{SN}$  (und somit die Unsicherheit  $U_{SN}$ ) in vier Teile zerlegt werden, Einflüsse der Iterationsanzahl  $\delta_I$ , Gittergröße  $\delta_G$ , Zeitschritt  $\delta_T$  und andere Parameter  $\delta_P$ . Damit ergibt sich folgender Ausdruck für die numerische Unsicherheit:

$$U_{SN}^2 = U_I^2 + U_G^2 + U_T^2 + U_P^2 \quad (19)$$

Im Validierungsprozess ist es notwendig einen Vergleichsfehler  $E$  zwischen den experimentellen Daten ( $D$ ) und dem Simulationsergebnis ( $S$ ) zu bestimmen.

$$E = D - S = \delta_D - (\delta_{SM} + \delta_{SN}) \quad (20)$$

Um feststellen zu können, ob ein Wert validiert ist, wird der Vergleichsfehler  $E$  mit der Validierungsunsicherheit  $U_V$  verglichen.

$$U_V^2 = U_D^2 + U_{SN}^2 \quad (21)$$

Falls  $|E| < U_V$ , ist die Kombination aller Fehler der experimentellen Daten und der Simulation kleiner als die Validierungsunsicherheit. Und das Simulationsergebnis ist validiert.[2] Falls  $U_V$  viel kleiner als  $|E|$  ist, kann der Fehler  $\delta_{SM}$  genutzt werden um Modellverbesserungen zu erzielen. Die korrigierten Simulationsergebnisse ergeben sich wie folgt:

$$E_C = D - S_C = \delta_D - (\delta_{SM} + \varepsilon_{SN}) \quad (22)$$

$$U_{VC}^2 = U_{EC}^2 - U_{SM}^2 = U_D^2 + U_{SN}^2 \quad (23)$$

## 6.1 Verifikationsverfahren

Im Rahmen der Verifikation muss eine Konvergenzstudie durchgeführt werden. Um diese Studie zu ermöglichen, sind mindestens drei kontinuierlich verfeinerte Gitter nötig. Zwischen diesen Gittern sollte ein gleichbleibender Verfeinerungsfaktor  $r_k$  existieren.

$$r_k = \frac{\Delta x_{k2}}{\Delta x_{k1}} = \frac{\Delta x_{k3}}{\Delta x_{k2}} = \frac{\Delta x_{km}}{\Delta x_{k_{m-1}}}$$

Dabei empfiehlt die ITTC einen Verfeinerungsfaktor der einzelnen Parameter (Zellenanzahl pro Koordinatenrichtung) von  $r_k = \sqrt{2}$ . Der Einfachheit halber wurde in dieser Arbeit  $r_k = 1,5$  gewählt, um leichter handhabbare Anzahlen von Zellen und ein ausreichend feines Gitter 1 zu erhalten.

Um die Art der Konvergenz oder die Divergenz einer Lösung zu zeigen wird der Konvergenzradius  $R_k$  definiert. Dafür sind mindestens drei verschiedene Gitter nötig. Der Konvergenzradius setzt sich aus den Änderungen ( $\varepsilon_{k_{ij}}$ ) der Lösungen ( $\hat{S}_{k_i}$  und  $\hat{S}_{k_j}$ ) zusammen. In dieser Arbeit sind drei Gitter vorhanden und somit setzt sich  $R_k$  wie folgt zusammen.

$$R_k = \frac{\varepsilon_{k21}}{\varepsilon_{k32}} \quad (24)$$

Dabei sind  $\varepsilon_{k21} = \hat{S}_{k2} - \hat{S}_{k1}$  und  $\varepsilon_{k32} = \hat{S}_{k3} - \hat{S}_{k2}$  die Änderungen zwischen benachbarten Lösungen.  $\hat{S}_{k1}$  ist die Lösung des feinen,  $\hat{S}_{k2}$  die des mittleren und  $\hat{S}_{k3}$  die des groben Gitters. Der Index  $k$  steht dabei für den analysierten Fehlereinfluss. [2]

Je nachdem welchen Wert  $R_k$  annimmt, gibt es drei mögliche Fälle:

$$\begin{aligned} (i) \quad & \text{monotone Konvergenz :} && 0 < R_k < 1 \\ (ii) \quad & \text{oszillierende Konvergenz :} && R_k < 0 \\ (iii) \quad & \text{Divergenz :} && R_k > 1 \end{aligned} \quad (25)$$

Für den Fall (i) wird die Generalisierte Richardson Extrapolation (RE) genutzt, um  $U_k$  oder  $\delta_k^*$  und  $U_{kC}$  zu bestimmen. Im Fall (ii) werden die Unsicherheiten abgeschätzt, indem der Fehler zwischen Oszillationsminima und -maxima bestimmt wird.

$$U_k = \frac{1}{2} (S_U - S_L) \quad (26)$$

Oszillierende Konvergenz kann fälschlicherweise als Fall (i) oder (iii) erkannt werden, je nachdem wo sich die untersuchten Punkte in der Oszillationskurve befinden. Daher ist es oft nur sinnvoll den Fehler  $\delta_k^*$  abzuschätzen, wenn mehr als drei Lösungen vorliegen. Für den Fall (iii) können Fehler und Unsicherheiten nicht bestimmt werden.

### Generalisierte Richardson Extrapolation

Für den Konvergenzfall (i) kann wie erwähnt die RE verwendet werden, um den Fehler  $\delta_k^*$  anhand vom Verfeinerungsfaktor und vom Genauigkeitsgrad  $P_k$  zu bestimmen.

Mit den vorhandenen drei Lösungen kann nur der Hauptterm des Fehlers bestimmt werden.

$$\delta_{RE_{k1}}^{*(1)} = \frac{\varepsilon_{k21}}{r_k^{P_k} - 1} \quad (27)$$

$$P_k = \frac{\ln(\varepsilon_{k32}/\varepsilon_{k21})}{\ln(r_k)} \quad (28)$$

### Abschätzung der Fehler und Unsicherheiten mit Korrekturfaktoren

Der Korrekturfaktor  $C_k$  wird verwendet, um die vorher vernachlässigten Terme höherer Ordnung mit in den Fehler  $\delta_{k1}^*$  einfließen zu lassen. Dieser korrigierte Fehler wird wie folgt definiert.

$$\delta_{k1}^* = C_k \cdot \delta_{RE_{k1}}^* = C_k \left( \frac{\varepsilon_{k21}}{r_k^{P_k} - 1} \right) \quad (29)$$

Für den Korrekturfaktor  $C_k$  wurden zwei verschiedene Ausdrücke entwickelt. Der erste basiert auf der Lösung der Gleichung (29) für  $C_k$  mit  $\delta_{RE_{k1}}^*$  aus der RE (Gl. 27). Er gibt nur sehr grob die Terme höherer Ordnung wieder.

$$C_k = \frac{r_k^{P_k} - 1}{r_k^{P_{kest}} - 1} \quad (30)$$

Der zweite Ausdruck für  $C_k$  ist eine genauere Abschätzung der höheren Fehlerterme. Dieser wird der Vollständigkeit halber wiedergegeben, in dieser Arbeit aber nicht verwendet.

$$C_k = \frac{\left( \frac{\varepsilon_{k23}}{\varepsilon_{k12}} - r_k^{q_{kest}} \right) (r_k^{P_k} - 1)}{(r_k^{P_{kest}} - r_k^{q_{kest}}) (r_k^{P_{kest}} - 1)} + \frac{\left( \frac{\varepsilon_{k23}}{\varepsilon_{k12}} - r_k^{p_{kest}} \right) (r_k^{P_k} - 1)}{(r_k^{p_{kest}} - r_k^{q_{kest}}) (r_k^{q_{kest}} - 1)}$$

Die Ausdrücke für Unsicherheiten werden anhand des korrigierten Fehlers (Gleichung (29)) entwickelt. Je nachdem welchen Wert der Korrekturfaktor annimmt, wird eine der folgenden

Gleichungen verwendet. Wenn  $C_k$  hinreichend kleiner oder größer als 1 ist, gilt folgende Gleichung

$$U_k = [|C_k| + |(1 - C_k)|] \cdot |\delta_{RE_{k1}}^*| \quad (31)$$

In den Quellwerken der ITTC UA in CFD [2] wird gezeigt, dass Gleichung (31) nicht konservativ genug für  $C_k < 1$  ist und daher wird für diesen Fall folgende Gleichung empfohlen.

$$U_k = [2 \cdot |1 - C_k| + 1] \cdot |\delta_{RE_{k1}}^*| \quad (32)$$

Wenn  $C_k$  nahe 1 ist, sodass der Fehler  $\delta_k^*$  abgeschätzt werden und die korrigierte Lösung  $S_C$  bestimmt werden kann, wird eine korrigierte Unsicherheit mit Gleichung (33) bestimmt. Dabei ist die einzige Einschränkung, dass  $C_k$  nicht 1 werden darf. In diesem Falle wird der Ansatz mit einem Sicherheitsfaktor empfohlen.

$$U_{kC} = |(1 - C_k)| \cdot |\delta_{RE_{k1}}^*| \quad (33)$$

### Abschätzung der Unsicherheiten mit Sicherheitsfaktoren

Wie oben kurz erwähnt, kann die Unsicherheit  $U_k$  auch mit einem Sicherheitsfaktor abgeschätzt werden.

$$U_k = F_S \cdot |\delta_{RE_{k1}}^*| \quad (34)$$

Dabei wird  $\delta_{RE_{k1}}^*$  mit der Gleichung (27) bestimmt. Der anzunehmende Wert des Sicherheitsfaktors ist nicht eindeutig bestimmbar. Die ITTC empfiehlt für eine sorgfältige Gitterstudie  $F_S = 1,25$  zu wählen. Für Studien mit nur zwei Gittern und angenommenem Genauigkeitsgrad  $p_{th}$  werden größere Werte bis  $F_S = 3$  nahe gelegt.

### Abschätzung der Fehler und Unsicherheiten für Punktvariablen

Das Bestimmen des Konvergenzradius  $R_k$  kann für Punktvariablen problematisch sein, denn  $\varepsilon_{k21}$  und  $\varepsilon_{k32}$  können beide Null werden. Der Konvergenzradius kann bei solchen Problemen trotzdem in bestimmten Regionen genutzt werden, zum Beispiel bei lokalen Maxima oder Minima.

Ein weiterer Ansatz ist die Nutzung eines globalen Konvergenzradius mit dem dazugehörigen Genauigkeitsgrad, der die erwähnten Probleme umgeht.

$$\langle R_k \rangle = \frac{\|\varepsilon_{k21}\|_2}{\|\varepsilon_{k32}\|_2} \quad (35)$$

$$\langle P_k \rangle = \frac{\ln \left( \frac{\|\varepsilon_{k21}\|_2}{\|\varepsilon_{k32}\|_2} \right)}{\ln(r_k)} \quad (36)$$

Dabei bezeichnen die Klammern  $\langle \rangle$  eine gemittelte Menge und die L2 Norm wird wie folgt definiert  $\|\varepsilon\|_2 = \left[ \sum_{i=1}^N \varepsilon_i^2 \right]^{\frac{1}{2}}$ .

Mit diesen Ausdrücken kann aber keine oszillierende Konvergenz bestimmt werden, daher ist vorsichtige Handhabung geboten.

## 6.2 Validierungsverfahren

Wenn der numerische Simulationsfehler stochastischer Art ist, wird die Unsicherheit  $U_{SN}$  abgeschätzt und es sind drei Variablen zu beachten ( $U_V$ ,  $|E|$  und  $U_{reqd}$ ). Die vorab definierte, einzuhaltende Simulationsunsicherheit  $U_{reqd}$  und die beiden zu berechnenden Unsicherheiten können in sechs verschiedenen Kombinationen zueinander stehen.

$$\begin{aligned}
 |E| &< U_V < U_{reqd} \\
 |E| &< U_{reqd} < U_V \\
 U_{reqd} &< |E| < U_V \\
 U_V &< |E| < U_{reqd} \\
 U_V &< U_{reqd} < |E| \\
 U_{reqd} &< U_V < |E|
 \end{aligned} \tag{37}$$

In den Fällen 1. - 3. (Gl. 37)) wird das Ergebnis validiert. Der Vergleichsfehler ist damit unter dem Störpegel, sodass die Abschätzung des Fehlers aus Modellannahmen  $\delta_{SMA}$  nicht möglich ist.

In den Fällen 4. - 6. (Gl. 37)) ist der Vergleichsfehler über dem Störpegel und mit dem entsprechenden Vorzeichen von  $E$  kann  $\delta_{SMA}$  bestimmt werden. Falls  $U_V \ll |E|$ , so ist  $E$  dem Fehler aus Modellannahmen zuzuordnen und  $\delta_{SMA}$  kann eindeutig bestimmt werden. Im Fall 4. ist Validierung aus Sicht der genutzten Software auf dem Niveau  $|E|$  erreicht.

Falls der numerische Simulationsfehler  $\delta_{SN}$  bestimmt werden kann, können auch  $\delta_{SN}^*$  und  $U_{V_C}$  abgeschätzt werden. Eine ähnliche Verhältnistabelle, wie im Falle eines stochastischen Fehlers, kann aufgebaut werden, indem man  $|E_C|$ ,  $U_{V_C}$  und  $U_{reqd}$  vergleicht.  $E_C$  kann größer oder kleiner als  $E$  sein, aber  $U_{V_C}$  muss immer kleiner als  $U_V$  sein. Die Ergebnisse müssen im korrigierten und unkorrigierten Fall nicht gleich sein.

## 6.3 Verifikation und Validierung - Gitter ohne FO

Die numerische Unsicherheit setzt sich aus vier Anteilen zusammen (siehe Gl. (19)). Die Unsicherheit des Zeitschritts  $U_T$  ist beim Gitter ohne FO nicht vorhanden, da die Rechnungen unter stationären Bedingungen ablaufen. Andere einfließende Parameter und somit deren Unsicherheiten  $U_P$  sind nicht bekannt. Daher vereinfacht sich die Gleichung (19) zu

$$U_{SN} = U_I + U_G \tag{38}$$

Bei den Rechnungen ohne FO wird der viskose Widerstandsbeiwert  $C_V$ , der Formfaktor  $k$  und die axiale Geschwindigkeit  $V_1$  (bei  $x/L=0,4911$  und  $z/L=-0,03$ ) verifiziert. Alle Daten ( $D$ ) der experimentell ermittelten Größen finden sich in Tabelle 13.

## viskoser Widerstandsbeiwert $C_V$

### Verifikation

Dieser Widerstandsanteil (errechnet aus der Gl. 14) wird (aus der Kräfteanalyse) mit der erwähnten OpenFOAM Funktion `forcesInter` bestimmt. Der zu validierende Simulationenswert  $S$  ist der des feinsten Gitters (siehe Tabelle 10)  $S_{C_V} = 3,29 \cdot 10^{-3}$ .

Die iterative Konvergenz ergibt sich bei Betrachtung des Verlaufs von  $C_V$  über die Iterationsanzahl. Der ausgegebene Wert schwankt dauerhaft leicht um einen Mittelwert und ist somit als oszillierend einzustufen. Um nun die iterative Unsicherheit zu bestimmen, wird der Kräfteverlauf von Gitter 1 (fein) analysiert [13]. Betrachtet wird die Abweichung vom Mittelwert über die letzten 4000 Iterationen (Iteration 6001 - 10000), da diese sich hier nicht mehr ändern (siehe Abb. 13). Die Unsicherheit wird über Oszillationsminima und -maxima bestimmt (siehe Gl. 26), woraus sich für  $C_V$  eine iterative Unsicherheit  $U_{I_{C_v}} = 0,01065 \cdot 10^{-3} = 0,3\% S_{C_V}$  ergibt.

Um die Gitterkonvergenz zu bestimmen, werden die Ergebnisse von  $C_V$  der drei Gitter (siehe Tab. 10) betrachtet. Bei Auswertung des Konvergenzradius der Lösungen (nach Gl. 24) ergibt sich der Fall (ii), oszillierende Konvergenz mit  $R_G = -10$ . Mit der Gleichung (26) wird die Gitterunsicherheit  $U_G = \frac{1}{2} \cdot (1,005 \cdot 3,44 \cdot 10^{-3} - 3,29 \cdot 10^{-3}) = 0,0836 \cdot 10^{-3} = 2,54\% S_{C_V}$  ermittelt. Dabei wurde die obere Grenze  $S_U$  0,5% größer als der höchste Gitterwert gewählt, da die Gitterwerte sehr wahrscheinlich nicht im Maximum oder Minimum der Oszillationskurve liegen.

Da die Fehlerterme  $\delta_G^*$  und  $\delta_I^*$  nicht bekannt sind, kann für  $C_V$  keine korrigierte Lösungen und Unsicherheit ermittelt werden.

Aus der vereinfachten Gleichung (19) ergibt sich die numerische Unsicherheit.

$$U_{S_{N_{C_v}}} = \sqrt{(0,0836 \cdot 10^{-3})^2 + (0,01065 \cdot 10^{-3})^2} = 0,0843 \cdot 10^{-3} = 2,56\% S_{C_V}$$

Die iterative Unsicherheit fällt fast nicht ins Gewicht. Das kann darauf hindeuten, dass ein noch feineres Gitter nötig ist, um den Gitterfehler zu verkleinern.

### Validierung

Die Validierung wird für den viskosen Widerstandsbeiwert mit dem Ergebnis des feinen Gitters  $S_{C_V} = 3,29 \cdot 10^{-3}$  durchgeführt. Zum Vergleich mit experimentellen Daten wird der entsprechende Wert der ITTC Korrelationslinie verwendet, mit einer Datenunsicherheit von  $U_{D_{C_v}} = 0,64\% D$ .

$$C_{FITTC} = \frac{0,075}{(\log(Rn) - 2)^2} = 2,832 \cdot 10^{-3} \quad (39)$$

Nun wird der Vergleichsfehler  $E$  über Gleichung (20) bestimmt.

$$E_{C_V} = D_{C_V} - S_{C_V} = 2,832 \cdot 10^{-3} - 3,29 \cdot 10^{-3} = -0,458 \cdot 10^{-3} = 16,2\% D_{C_V}$$

Die Validierungsunsicherheit wird aus Gleichung (21) berechnet.

$$U_{V_{C_v}} = \sqrt{U_{S_{N_{C_v}}}^2 + U_{D_{C_v}}^2} = 0,0862 \cdot 10^{-3} = 3,04\% D_{C_V}$$

Man sieht sofort, dass der Vergleichsfehler  $|E_{C_V}| > U_{V_{C_V}}$ . Damit sind die *Simulationsergebnisse für  $C_V$  nicht validiert*.

$E_{C_V}$  ist negativ und betragsmäßig groß, d.h. die Simulationssoftware errechnet mit den verwendeten Einstellungen, bei dieser Gittergröße ohne FO (ca. 300000 Zellen) überzogene Reibungskräfte.

## **Formfaktor $k$**

### Verifikation

Der Formfaktor  $k$  wurde mittels Gleichung (13) bestimmt und die Ergebnisse der Gitter sind in Tabelle 10 aufgelistet. Der zu validierende Simulationswert des feinsten Gitters ist  $S_k = 22, 11$ .

Die iterative Konvergenz wurde bereits oben (Validierung von  $C_V$ ) bestimmt. Diese leistet keinen entscheidenden Beitrag zur numerischen Unsicherheit, daher wird sie hier vernachlässigt.

Die Gitterkonvergenz wird über den Konvergenzradius  $R_k$  bestimmt.

$$R_k = \frac{\varepsilon_{k_{21}}}{\varepsilon_{k_{32}}} = 11, 29$$

Damit ergibt sich ein divergentes Verhalten des Formfaktors in Bezug auf die möglichen Fälle der Gleichung (25). Gitterfehler und -unsicherheiten können bei Divergenz nicht bestimmt werden.

### Validierung

Der Formfaktor des KCS wird nicht auf der Webseite des Gothenburg 2000 Workshops [1] angegeben. In einem Paper zu den Experimenten des KCS, die vor dem Workshop durchgeführt wurden, wird der Formeinfluss mit ca. 10% angegeben. Dies ist weit unter dem Wert des Gitters 1 und lässt darauf schließen, dass OpenFOAM bei dieser Zellenanzahl die Druckkräfte im Vergleich zu den Reibungskräften überschätzt.

Wie oben erwähnt, machen die Gitterunsicherheiten fast die gesamte numerische Unsicherheit  $U_{SN}$  aus. Da  $k$  aber nicht über  $R_k$  konvergiert, kann  $U_{G_k}$  nicht bestimmt werden und somit werden die *Ergebnisse für den Formfaktor nicht validiert*.

## **axiale Geschwindigkeit $V1$ bei $x/L=0,4911$ und $z/L=-0,03$**

### Verifikation

Die Ergebnisse der normierten Nachstromgeschwindigkeit in x-Richtung ( $V1$ ) sind in Abb. 7 zu sehen.  $V1$  ist eine Punktvariable, d.h. sie ist an bestimmten Punkten des Gitters definiert. Da die Gitter 1-3 unterschiedlich sind, ist  $V1$  dementsprechend an unterschiedlichen Punkten in jedem Gitter definiert. Um Ergebnisse zu erhalten, die mit den experimentellen Daten vergleichbar sind, muss zwischen den Punkten interpoliert werden. Daher wurde linear interpoliert, um Simulationswerte in den 22 experimentell ermittelten Punkten zu erhalten.

Wie bei der der Verifikation des viskosen Widerstandsbeiwerts  $C_V$  gezeigt wurde, sind die iterativen Fehler und Unsicherheiten viel kleiner als die Gitterfehler und Unsicherheiten ( $\delta_I \ll \delta_G$  und  $U_I \ll U_G$ ), sodass  $U_{SN} = U_G$ .

Der globale Konvergenzradius  $\langle R_{G_{V1}} \rangle = 0,776$  konvergiert monoton. Bei Untersuchung jedes einzelnen Punktes (Ergebnisse siehe Tabelle 17 und Abb. 14) sieht man, dass nur 3 von 22 Punkten nicht konvergieren, alle anderen entweder monoton oder oszillierend. Der Genauigkeitsgrad wird über die L2 Norm bestimmt (siehe Gl. 36).

$$\langle P_{G_{V1}} \rangle = \frac{\ln \left( \frac{\|\varepsilon_{G21}\|_2}{\|\varepsilon_{G32}\|_2} \right)}{\ln(r_G)} = 0,625$$

Da  $V1$  nicht an allen Punkten konvergiert, wurde die Abweichung der einzelnen Gitter in Bezug auf das Experiment in Graph 15 dargestellt. Dies soll zeigen, dass sich die Lösung des feinen Gitters deutlich an das Experiment annähert.

Der globale Korrekturfaktor  $\langle C_{G_{V1}} \rangle$  wird nachfolgend bestimmt, wobei der benötigte Wert der geschätzten Genauigkeitsordnung  $p_{est} = p_{th} = 2$  aus dem IIHR Report 407 (von Stern, 1999 [13]) übernommen wird.

$$\langle C_{G_{V1}} \rangle = \frac{r_G^{P_{G_{V1}}} - 1}{r_G^{P_{est}} - 1} = 0,231$$

Nun werden die Fehler und Unsicherheiten bestimmt. Wenn  $\langle C_{G_{V1}} \rangle = 0,231$  als "ausreichend kleiner" als 1 beurteilt wird, kann die resultierende Unsicherheit  $U_{G_{V1}}$  über Gleichung (32) für jeden Punkt bestimmt werden, nicht aber der Gitterfehler  $\delta_{G_{V1}}$  [13].

$$U_{G_{V1}} = \left| \langle C_{G_{V1}} \rangle \cdot \frac{\varepsilon_{G21}}{r_G^{P_{G_{V1}}} - 1} \right| + \left| (1 - \langle C_{G_{V1}} \rangle) \cdot \frac{\varepsilon_{G21}}{r_G^{P_{G_{V1}}} - 1} \right|$$

Wenn hingegen  $\langle C_{G_{V1}} \rangle = 0,231$  als "dicht bei" 1 beurteilt wird, kann für jeden Punkt eine Fehlerabschätzung  $\delta_{G_{V1}}^*$  und die korrigierte Unsicherheit  $U_{G.C_{V1}}$  ermittelt werden.

$$\delta_{G_{V1}}^* = \langle C_{G_{V1}} \rangle \left( \frac{\varepsilon_{G21}}{r_G^{P_{G_{V1}}} - 1} \right)$$

$$U_{G.C_{V1}} = \left| (1 - \langle C_{G_{V1}} \rangle) \left( \frac{\varepsilon_{G21}}{r_G^{P_{G_{V1}}} - 1} \right) \right|$$

Mit der Fehlerabschätzung  $\delta_{G_{V1}}^*$  kann zusätzlich noch das korrigierte Simulationsergebnis  $S_{C_{V1}}$  bestimmt werden.

$$S_{C_{V1}} = V1_{fine} - \delta_{G_{V1}}$$

Alle Ergebnisse aller 22 Punkte sind in Tabelle 18 aufgeführt.

### Validierung

Die Validierung von  $V1$  wird jeweils mit dem Simulationsergebnis  $S_{V1} = V1_{fein}$  und dem

korrigierten Simulationsergebnis  $S_{C_{V1}} = V1_{Exp} - \delta_{G_{V1}}^*$  durchgeführt. Die Ergebnisse sind im Anhang in Tabelle 18 zu sehen. Die Datenunsicherheit der Experimente  $U_D = 0,38\% \cdot D_{V1}$  wurde von der KRISO Website [1] übernommen.

Der Vergleichsfehler  $E_{V1} = D_{V1} - S_{V1}$  wird mit der Validierungsunsicherheit  $U_{V1} = \sqrt{U_D^2 + U_{G_{V1}}^2}$  ausgewertet. Der korrigierte Vergleichsfehler  $E_{V1C} = D_{V1} - S_{C_{V1}}$  wird entsprechend mit der korrigierten Validierungsunsicherheit  $U_{VC_{V1}} = \sqrt{U_D^2 + U_{G.C_{V1}}^2}$  ausgewertet. Die *Ergebnisse können annähernd vollständig validiert werden*, bis auf den Punkt Nr 1 und die drei Punkte mit divergierenden Konvergenzradien  $R_{G_{V1}} > 1$  (Tab. 18).

Das Simulationsergebnis ist teilweise sehr dicht am Experiment und der Vergleichsfehler  $E$  befindet sich nahe Null (siehe Abb. 16). Auffällig und unerwünscht ist aber, dass die Validierungsunsicherheit  $U_V$  teilweise 80 % -100 % vom Simulationsergebnis  $S_{V1}$  erreicht. Das geschieht, weil die Ergebnisse des mittleren und groben Gitters viel deutlicher vom Experiment abweichen. Mit einem noch weiter verfeinerten Gitter und somit genauerem Ergebnis, könnte man die Validierungsunsicherheit wahrscheinlich deutlich verringern. Die Korrektur des Simulationsergebnisses bringt zwar etwas kleinere korrigierte Unsicherheiten mit sich (siehe Tab. 18 und Abb. 17), dafür wird aber das korrigierte Ergebnis im vorher sehr genauen Bereich ( $y/L = 0,01$  bis  $0,02$ ) verschlechtert.

## 6.4 Verifikation und Validierung - Gitter mit FO

Genau wie beim Gitter ohne FO setzt sich die numerische Unsicherheit vereinfacht aus iterativer und Gitterunsicherheit zusammen (Gl. (38)). Der Unterschied ist die hier vorhandene reflektierende Welle, die der iterativen Unsicherheit große Bedeutung gibt.

Bei den Rechnungen mit FO wird der Gesamtwiderstandsbeiwert  $C_T$ , der Restwiderstandsbeiwert  $C_R$  und die Wellenhöhe  $\frac{z}{L}$  im Wellenlängsschnitt (bei  $y/L=0,1509$ ) verifiziert.

### Gesamtwiderstandsbeiwert $C_T$

#### Verifikation

Die iterative Konvergenz wird ermittelt, indem man den Gesamtkraftverlauf und den daraus resultierenden Gesamtwiderstandsbeiwert betrachtet (siehe Abb. 18). Der  $C_T$  Wert oszilliert mit abnehmender Amplitude und konvergiert langsam gegen einen Mittelwert. Die in der letzten Oszillation registrierten maximalen und minimalen Auslenkungen ( $C_{T_{max}} = 4,583 \cdot 10^{-3}$  und  $C_{T_{min}} = 3,769 \cdot 10^{-3}$ ) werden in die Gleichung für oszillierende Konvergenz (Gl. (26)) eingesetzt. Damit ergibt sich eine iterative Unsicherheit von  $U_{I_{Ct}} = 0.407 \cdot 10^{-3}$  was 9,72% von  $S_{C_T}$  entspricht.

Um die Gitterkonvergenz nachzuweisen, wird der Konvergenzradius der Lösungen der drei Gitter (Werte siehe Tab. 11) bestimmt. Es ergibt sich  $R_{G_{Ct}} = 0,293$  und somit Fall (ii), monotone Konvergenz (siehe Gl. 25). Nun kann der Fehler  $\delta_{RE.G_{Ct}}^*$  sowie der Genauigkeits-

grad  $P_{G_{Ct}}$  über RE und der Korrekturfaktor  $C_{G_{Ct}}$  bestimmt werden.

$$\delta_{RE.G_{Ct}}^* = \frac{\varepsilon_{G_{21}}}{r_G^{P_G} - 1} = \frac{0,286}{1,5^{3,027} - 1} = 0,1186 \cdot 10^{-3}$$

$$P_{G_{Ct}} = \frac{\ln(\varepsilon_{G_{32}}/\varepsilon_{G_{21}})}{\ln(r_G)} = \frac{\ln\left(\frac{0,9762}{0,286}\right)}{\ln(1,5)} = 3,027$$

$$C_{G_{Ct}} = \frac{r_G^{P_G} - 1}{r_G^{P_{kest}} - 1} = \frac{1,5^{3,027} - 1}{1,5^2 - 1} = 1,93$$

Wie bei der Verifikation der normierten Nachstromgeschwindigkeit  $V1$  wurde  $p_{est} = p_{th} = 2$  benutzt. Nun wird die Gitterunsicherheit zweimal bestimmt. Einmal unter der Annahme  $C_{G_{Ct}} = 1,93$  ist hinreichend größer als 1 und ein zweites Mal mit  $C_{G_{Ct}} = 1,93$  ist nahe 1. Für  $C_{G_{Ct}} = 1,93$  hinreichend größer 1 kann nur die Gitterunsicherheit über Gleichung (31) abgeschätzt werden.

$$U_{G_{Ct}} = [|C_{G_{Ct}}| + |(1 - C_{G_{Ct}})|] \cdot |\delta_{RE.G_{Ct}}^*| = 1,93 \cdot |0,1186 \cdot 10^{-3}| = 0,339 \cdot 10^{-3}$$

Damit ist  $U_{G_{Ct}} = 0,339 \cdot 10^{-3}$  gleich 8,1% von  $S_{C_T}$ .

Für  $C_{G_{Ct}} = 1,93$  erachtet als nahe 1, kann eine Fehlerabschätzung  $\delta_{G_{Ct}}^*$  (Gl. (32)) und der korrigierte Gitterfehler  $U_{G.C_{Ct}}$  (Gl. (29)) bestimmt werden.

$$\delta_{G_{Ct}}^* = C_{G_{Ct}} \cdot \delta_{RE.G_{Ct}}^* = 1,93 \cdot 0,1186 \cdot 10^{-3} = 0,229 \cdot 10^{-3}$$

$$U_{G.C_{Ct}} = |(1 - C_{G_{Ct}}) \cdot \delta_{RE.G_{Ct}}^*| = |(1 - 1,93) \cdot 0,1186 \cdot 10^{-3}| = 0,11 \cdot 10^{-3}$$

Damit kann die Lösung korrigiert werden.

$$S_{C_T.C} = S_{C_T} - \delta_{G_{Ct}}^* = 3,958 \cdot 10^{-3}$$

Die unkorrigierte Unsicherheit  $U_{G_{Ct}} = 8,1\% S_{C_T}$  ist vergleichsweise hoch. Dafür halten sich die korrigierte Unsicherheit mit  $U_{G.C_{Ct}} = 2,63\% S_{C_T}$  sowie der Fehler  $\delta_{G_{Ct}}^* = 5,47\% S_{C_T}$  in einem guten Rahmen.

Mit den gewonnenen Daten können die numerische Unsicherheit und die korrigierte numerische Unsicherheit für  $C_T$  über Gleichung (19) berechnet werden.

$$U_{SN_{Ct}} = \sqrt{U_{G_{Ct}}^2 + U_{I_{Ct}}^2} = 0,53 \cdot 10^{-3} = 14,9\% D_{D_T}$$

$$U_{SN.C_{Ct}} = \sqrt{U_{G.C_{Ct}}^2 + U_{I_{Ct}}^2} = 0,4217 \cdot 10^{-3} = 11,85\% D_{D_T}$$

### Validierung

Die Validierung wird einmal mit dem Simulationsergebnis  $S_{C_T}$  und dem korrigierten Simulationsergebnis  $S_{C_T.C}$  durchgeführt.

Als erstes wird der Fehler  $E_{C_T}$  mit dem Simulationsergebnis und dem experimentellen Wert  $D_{C_T} = 3,56 \cdot 10^{-3}$  bestimmt.

$$E_{C_T} = D_{C_T} - S_{C_T} = 3,56 \cdot 10^{-3} - 4,187 \cdot 10^{-3} = 0,627 \cdot 10^{-3} = 17,61\% D_{C_T}$$

Die Validierungsunsicherheit wird aus Gleichung (21) berechnet.

$$U_{V_{C_t}} = \sqrt{U_{S_{N_{C_t}}}^2 + U_{D_{C_t}}^2} = 0,53 \cdot 10^{-3} = 14,9\% D_{D_T}$$

Damit ist der Vergleichsfehler  $|E_{C_T}| > U_{V_{C_t}}$  und das *Simulationsergebnis wird nicht validiert*.

Um die Validierungsunsicherheit zu verkleinern, müsste die Simulationsunsicherheit verkleinert werden. Die Gitterunsicherheit und die iterative Unsicherheit sind von gleicher Ordnung, sodass man beide verkleinern müsste. Dies könnte mit einem feineren Gitter erreicht werden, obwohl die reflektierende Welle (und somit  $U_I$ ) dadurch nicht verringert wird. Der Vergleichsfehler  $E$  ist negativ, somit überschätzt die Simulation die vorhandenen Kräfte.

Nun werden die korrigierten Simulationsergebnisse betrachtet. Der korrigierte Vergleichsfehler wird wie folgt bestimmt.

$$E_{C_T.C} = D_{C_T} - S_{C_T.C} = 3,56 \cdot 10^{-3} - 3,958 \cdot 10^{-3} = 0,398 \cdot 10^{-3} = 11,19\% D_{C_T}$$

Die korrigierte Validierungsunsicherheit ergibt sich zu

$$U_{V_{C_{C_t}}} = \sqrt{U_{S_{N_{C_{C_t}}}}^2 + U_{D_{C_t}}^2} = 0,4223 \cdot 10^{-3} = 11,86\% D_{D_T}$$

Der korrigierte Vergleichsfehler ist kleiner als die korrigierte Validierungsunsicherheit  $|E_{C_T.C}| < U_{V_{C_{C_t}}}$ . Damit ist das *korrigierte Simulationsergebnis von  $C_T$  validiert*.

Die Validierungsunsicherheit ist mit  $11,86\% D_{D_T}$  zwar recht hoch, aber für die Anzahl an Zellen, den geringen Erfahrungszeitraum mit OpenFOAM meinerseits, sowie die dürftige existente Dokumentation durchaus vertretbar.

### **Restwiderstandsbeiwert $C_R$**

#### Verifikation

$C_R$  wird mit Gleichung (16) berechnet. Es wird mit der Gitterkonvergenz begonnen. Man betrachte die Ergebnisse in Tabelle 11. Der Wert von  $C_{R_3}$  (grobes Gitter) ist viel höher als die anderen beiden. Man kann nun einen Konvergenzradius  $R_{G_{C_r}} = -0,036$  bestimmen, der auf oszillierende Konvergenz hinweist. Wenn man aber die Gitterunsicherheit über Gleichung (26) bestimmt, ergibt sich  $U_{G_{C_r}} = 0,51 \cdot 10^{-3}$ , was einer Unsicherheit von fast 100% gegenüber dem Simulationsergebnis entspricht. Mit so einer hohen Unsicherheit ist es

nicht sinnvoll eine Validierung durchzuführen.

Die Ursache dafür liegt darin, dass der Reibungsbeiwert  $C_F$  bei feiner werdendem Gitter nicht abnimmt, der Druckbeiwert  $C_P$  seinerseits aber ziemlich stark. Daher ergibt sich auch der hohe Restwiderstandsbeiwert beim groben Gitter. Um dies zu verhindern, müsste man die reflektierende Welle in OpenFOAM eliminieren. Die Welle ist aber ein bekanntes Problem und konnte bisher von keiner mir bekannten Quelle (Person/Institution) eliminiert werden.

#### Validierung

Aus oben genannten Gründen wird  $C_R$  *nicht validiert*.

### **Wellenhöhe $\frac{z}{L}$ im Wellenlängsschnitt bei $y/L=0,1509$**

#### Verifikation

Das Wellenprofil wurde in einem Moment aufgenommen, in dem die gemessenen Kräfte dem Mittelwert über eine Schwingungsperiode entsprechen. Bei OpenFOAM kann man sich die Gesamtkräfte per `forcesInter` für jeden Iterationsschritt in einer Datei ausgeben lassen, nicht aber eine Wellenhöhe oder Geschwindigkeit an einem bestimmtem Punkt, ohne riesige Datenmengen zu erzeugen. Daher ist es kompliziert in diesem Fall die iterative Ungenauigkeit zu bestimmen. Um weitere Analysen der Wellenhöhe vornehmen zu können, wird davon ausgegangen, dass nach unendlich vielen Iterationen das Wellenbild dem gemessenen Mittel entspricht und somit  $\delta_{I_{z/L}} \ll \delta_{G_{z/L}}$  sowie  $U_{I_{z/L}} \ll U_{G_{z/L}}$ .

Die Wellenhöhen der drei Gitter und des Experiments sind in Abbildung 10 zu sehen. Das Experiment liefert Werte in 500 Punkten zwischen  $-0,5 \leq x/L \leq 1$ . Da im Experiment mit grobem Gitter nur weniger als 120 Werte zur Verfügung stehen, wurde bei allen Gittern und dem Experiment in 150 gleichmäßig verteilten Punkten linear interpoliert.

Die Gitterkonvergenz wird über den globalen Konvergenzradius wie in Gleichung (35) bestimmt,  $R_{G_{z/L}} = 2,5$ . Da  $R_{G_{z/L}} > 1$  ist, divergiert der globale Konvergenzradius des Gitters. Es wurde zusätzlich der punktuelle Konvergenzradius in den 150 betrachteten Punkten ermittelt (siehe Abb. 19). Man sieht, dass im Bereich der Bugwelle (ca.  $-0,5 \leq x/L < -0,2$ )  $R_{G_{z/L}}$  in jedem Punkt divergiert. Das geschieht, da die Wellenhöhe beim mittleren Gitter im Bugbereich sehr gut mit dem Experiment übereinstimmt und das feine Gitter die Bugwelle überschätzt (siehe Abb. 10).

Nach dem Bugbereich passt sich das Simulationsergebnis immer besser dem Experiment an und es ergibt sich im Großteil der Punkte ein konvergierender  $R_{G_{z/L}}$ . Man könnte nun den Genauigkeitsgrad  $P_{G_{z/L}} = -2,26$  bestimmen, der negativ ausfällt. Da aber auch nach der Bugwelle immer wieder Punkte einen Konvergenzradius größer 1 haben, wird die Gitterunicherheit  $U_{G_{z/L}}$  als nicht bestimmbar erklärt.

#### Validierung

Aus oben genannten Gründen wird die Wellenhöhe  $\frac{z}{L}$  im Wellenlängsschnitt *nicht validiert*.

*Eine tabellarische Übersicht aller Validierungsergebnisse findet sich im Anhang, siehe Tabelle 16.*

## 7 Vergleich der Ergebnisse mit anderen Arbeiten

Während des CFD-Workshops Gothenburg 2000 wurden drei Modellschiffsrümpfe (KVLCC, KCS und Modell 5415) von 20 Teilnehmergruppen untersucht. Unter den Teilnehmern befanden sich 16 Institute (akademische und industrielle), von denen drei Institute zweifach Untersuchungen durchführten, und eine kommerzielle CFD Firma. Von diesen Teilnehmern haben sieben Gruppen CFD-Rechnungen zum KRISO Containerschiff durchgeführt. Vom KCS wurden unter anderem der Gesamtwiderstandsbeiwert  $C_T$ , mit seinen Komponenten  $C_F$  und  $C_P$ , sowie der Wellenschnitt bei  $y/L = 0,1509$  und die Geschwindigkeit  $V_1$  bei  $x/L = 0,4911$  und bei  $z/L = -0,03$  untersucht. Die Ergebnisse des Workshops sind ausführlich zusammengefasst in den Gothenburg 2000 Proceedings [8].

Es wurden teilweise Simulationsergebnisse erzielt, die sehr nah an den experimentellen Daten liegen, siehe Ergebnisse der Widerstandsbeiwerte in Tabelle 19 im Anhang. Dabei variiert die Gitterauflösung der Workshopteilnehmer zwischen 0,4 und 1,0 Millionen Zellen.

### Vergleich von $C_T$

Der Durchschnittswert der Workshopteilnehmer für  $C_T$  ist 5,2 % höher als der experimentell gemessene [8]. Der größte Vergleichsfehler  $E$  beträgt 10,67% und der kleinste nur 0,28% Abweichung vom Experiment. Es wurden nur von zwei Teilnehmern nachvollziehbare Validierungsprozeduren durchgeführt um  $U_V$  zu ermitteln (siehe Tab. 19), wovon ein Ergebnis mit 4,1%  $C_T$  validiert werden konnte. Im Vergleich ergibt die OpenFOAM Rechnung 17,6% Abweichung und das korrigierte Simulationsergebnis 11,2% Abweichung vom Experiment. Das korrigierte Simulationsergebnis konnte im Kapitel 6.4 mit einer recht hohen Validierungsunsicherheit ( $U_{V,C} = 11,86\%$ ) validiert werden.

Die Vorhersagen der Workshopteilnehmer für  $C_F$  variieren schon stärker und die Ergebnisse für  $C_P$  weichen im Schnitt sogar über 35% vom Mittelwert aller Teilnehmer ab. Dabei kommen Werte für  $C_P$  zustande die zwischen  $C_P = 0,37$  und  $1,36$  liegen. Daraus ist ersichtlich, dass mit guter Kenntnis der eigenen Simulationssoftware und des Turbulenzmodells, passende Ergebnisse für eine wichtige gesuchte Variable erzielt werden können (hier  $C_T$ ). Einzelne Komponenten dieser können aber trotzdem große Fehler enthalten, obwohl das Endergebnis stimmt. Man sieht dabei, dass die Workshopteilnehmer ihre CFD-Codes sehr gut kennen und wissen, wie man nah an einen wichtigen gesuchten Wert kommt.

Die durchschnittliche numerische Unsicherheit während der Verifikation lag bei 2,3%. Die durchgeführte OpenFOAM Rechnung hat erheblich höhere Werte für  $U_{SN} = 14,9\%$  und  $U_{SN_C} = 11,85\%$ . Diese Tatsache ist hauptsächlich der iterativen Unsicherheit ( $U_I = 9,72\%$ ) und somit der reflektierenden Welle zuzuordnen.

Man kann in die Betrachtung auch zusätzlich die Ergebnisse des Workshops für ein weiteres Modell mit einbeziehen. Zwischen den Modellen KCS und 5415 wurden in Gothenburg 2000 fünf Codes validiert, mit Niveaus zwischen 3 – 15%  $C_T$ . Der durchschnittliche Vergleichsfehler  $E$  für  $C_T$  lag bei 4,8%. Damit liegt das korrigierte Simulationsergebnis in OpenFOAM  $C_{T_C} = 11,19\% D_{C_T}$ , validiert bei 11,8%  $C_T$ , in einem vertretbaren Bereich.

### **Vergleich vom Wellenprofil/Wellenschnitt**

Das Wellenfeld um den Rumpf wurde von allen Workshopteilnehmern gut vorhergesagt, mit Ausnahme der Bugwelle, die meist unterschätzt wird. Bei den vorliegenden Rechnungen mit OpenFOAM wird die Bugwelle beim feinsten Gitter im Gegensatz dazu etwas überschätzt. Die Heckwelle (bei  $x/L = 0,5$ ) wird von fast allen Teilnehmern (mit einer Ausnahme) zu groß prognostiziert, was bei den OpenFOAM Rechnungen genauso ist (siehe Abb. 9).

In einiger Entfernung vom Rumpf waren die Vorhersagen der meisten Workshopteilnehmer nicht mehr ausreichend genau (mit den Ausnahmen: HSVA-NEPTUN und MARIN/IST-PARNASSUS)[8]. Die Wellen sind zu gedämpft am Rande des aufgelösten Gebiets, was wahrscheinlich an Auflösungsproblemen liegt. Das in OpenFOAM bestimmte Wellenfeld stimmt optisch sehr gut mit dem Experiment überein (Abb. 9). Der Rand des Lösungsgebiets ist auch gut aufgelöst. Ab ca.  $x/L=1$ , also eine halbe Schiffslänge hinter dem Rumpf, werden die Wellen natürlich nicht mehr optimal aufgelöst, da die Zellenabstände größer werden, um Reflexionen kleiner zu halten.

Der Wellenschnitt bei  $y/L = 0,1509$  wird von den Workshopteilnehmern auch gut wiedergegeben, mit 2 Ausnahmen. Die zweite Komponente der Mittschiffswelle wird aber von keiner viskosen Methoden erfasst, nur die Potenzialmethode von MARIN/IST-PARNASSUS gibt diese wieder. Die Wellenhöhe bei  $x/L = 1$  ist um einiges kleiner als beim Experiment, was eindeutig auf Auflösungsprobleme hindeutet. Im Vergleich gibt die OpenFOAM-Rechnung den Wellenschnitt auch gut wieder, mit dem gleichen Problem der viskosen Codes, dass die Mittschiffswelle nicht abgestuft ist (siehe Abb. 10). Die Wellenhöhe bei  $x/L = 1$  ist auch kleiner als beim Experiment, da die Zellen des Gitters ab  $x/L = 0,618$  schrittweise größer werden. Dabei ist nochmal zu erwähnen, dass die Ergebnisse des Wellenschnitts des mittleren Gitters an Bug und Heck des Rumpfes überraschend genau mit dem Experiment übereinstimmen.

### **Vergleich der Nachstromgeschwindigkeit**

Das Experiment ergibt einen Nachstromwirbel, der seinen Mittelpunkt bei  $z/L = -0,025$  hat. Der axiale Geschwindigkeitsschnitt bei  $z/L = -0,03$  weist einen plateauähnlichen Verlauf auf, was durch den Wirbel verursacht wird. Fast alle Workshopteilnehmer können grob beide Eigenschaften nachweisen, obwohl die Stärke des Wirbels und die Größe des Plateaus etwas variieren. Allgemein gibt es aber die Tendenz, den Nachstrom zu schwach vorherzusagen und die Querströmung nicht gut genug aufzulösen.

In den Ergebnissen der OpenFOAM-Rechnung ist der Wirbel nicht stark genug ausgeprägt, die Geschwindigkeitsverteilung ist dem Experiment aber sehr ähnlich (siehe Abb. 6). Allgemein ist die  $z$ -Komponente der Geschwindigkeit ( $w$ ) etwas schwächer als im Experiment, genau wie bei den Workshopergebnissen.

Die Geschwindigkeitsverteilung beim Schnitt auf Höhe  $x/L = -0,03$  wird von verschiedenen Workshopgruppen ordentlich wiedergegeben, zumindest ist die Tendenz bei allen zu sehen. Die OpenFOAM-Rechnung gibt den Geschwindigkeitsverlauf von  $V1$  auch gut wieder (siehe

Abb. 7 links), bis auf das Gebiet sehr Nahe  $y/L = 0$ . Wie im Abschnitt der Validierung zu lesen ist, wurde  $V1$  fast komplett validiert.

## **Paper zum Gotheburg 2000 Workshop von Cura, A. & Vogt, M. [4]**

Diese Veröffentlichung mit dem Namen "Flow and Resistance Prediction for a Container Ship" wurde von den Autoren zusätzlich zu den Ergebnissen im CFD-Workshop Gothenburg 2000 eingereicht. Darin wurden der Widerstand, das Wellenfeld und zwei zusätzliche Schnitte des Wellenfeldes vom KCS in Modellgröße ausgewertet.

Für die Simulation wurde der von den Autoren entwickelte RANS-Code Neptun verwendet. Die Turbulenz wird mit dem  $k - \omega$  Turbulenzmodell erfasst. Dabei wurden Wandfunktionen verwendet. Die Position der freien Oberfläche wurde mit der Level Set Methode bestimmt. Die Erhaltungsgleichungen sind mit einer Finite-Volumen Methode diskretisiert. Beim Lösen der Gleichungen werden die Diffusionsterme mit CDS (Central Differencing Scheme) und die konvektiven Geschwindigkeiten mit LUDS (Linear Upwind Differencing Scheme) approximiert. Zusätzlich wurde der SIMPLE Algorithmus genutzt. Das feinste Gitter hatte 672000 Zellen, wobei der Einlass 3 L (Schiffslängen) vor dem Schiff lag und der Auslass 5 L dahinter. Die Breite des Gitters ist 3 L und die Wassertiefe 1 L.

Die Ergebnisse des feinen Gitters sind  $C_F = 2,96 \cdot 10^{-3}$  und  $C_T = 3,93 \cdot 10^{-3}$ . Damit wird  $C_{FITC}$  um 4,6 % und  $C_T$  um ca. 10 % überschätzt. Die Autoren weisen darauf hin, dass ein feineres Gitter bessere Ergebnisse liefern würde. Mit OpenFOAM wurden ähnlich überhöhte Werte für den Gesamtwiderstandsbeiwert ermittelt. Der Wert für  $C_T$  konnte aber nach einer Korrektur validiert werden.

Das globale Wellenbild wird von Cura/Vogt sehr exakt ausgegeben (siehe Abb. 11) und ist auch das beste Ergebnis aller Teilnehmer im Workshop Gothenburg 2000. Wenn man das Wellenbild aus OpenFOAM (Abb. 9) mit diesem vergleicht sieht man, dass das Cura/Vogt Wellenbild im interessanten Bereich etwas genauer mit dem Experiment überein stimmt. Vor allem der Wellenberg mittschiffs und die Heckwelle werden sehr gut aufgelöst. Der Bereich hinter dem Schiff wird dann nicht mehr so gut erfasst, wahrscheinlich weil die Zellen dort sehr grob werden. Bei der OpenFOAM Simulation wird der Hinterschiffsbereich ein wenig genauer aufgelöst und so hat man hinter dem Schiff noch ein angemessenes Wellenbild.

Die zusätzlichen Wellenschnitte im Paper, bei  $y/L = 0,1024$  und  $y/L = 0,3089$  wurden in dieser Bachelorarbeit nicht ausgewertet.

Die getroffenen Annahmen und die Gittereinstellungen von Cura/Vogt sind ähnlich denen in dieser Arbeit. Mit Ausnahme, dass in den OpenFOAM Rechnungen das  $k - \omega - SST$  Modell verwendet und die freie Oberfläche mit der VOF-Methode bestimmt wird. Sogar die verwendete Zellanzahl ist fast gleich groß. Die Ergebnisse sind insgesamt sehr ähnlich, wenn man die auswertbaren Größen betrachtet. Man muss aber insgesamt von einer CFD-Rechnung, die 10 Jahre später erstellt wurde erwarten, dass sie Verbesserungen bringt.

Da aber die Zellanzahl gleich ist und in Sachen Turbulenzmodellierung keine Fortschritte im letzten Jahrzehnt erzielt wurden, ist das Ergebnis realistisch. Vor allem unter dem Gesichtspunkt, dass Cura/Vogt ihre CFD-Software Neptun selbst entwickelt haben und somit alle ihre Feinheiten kennen.

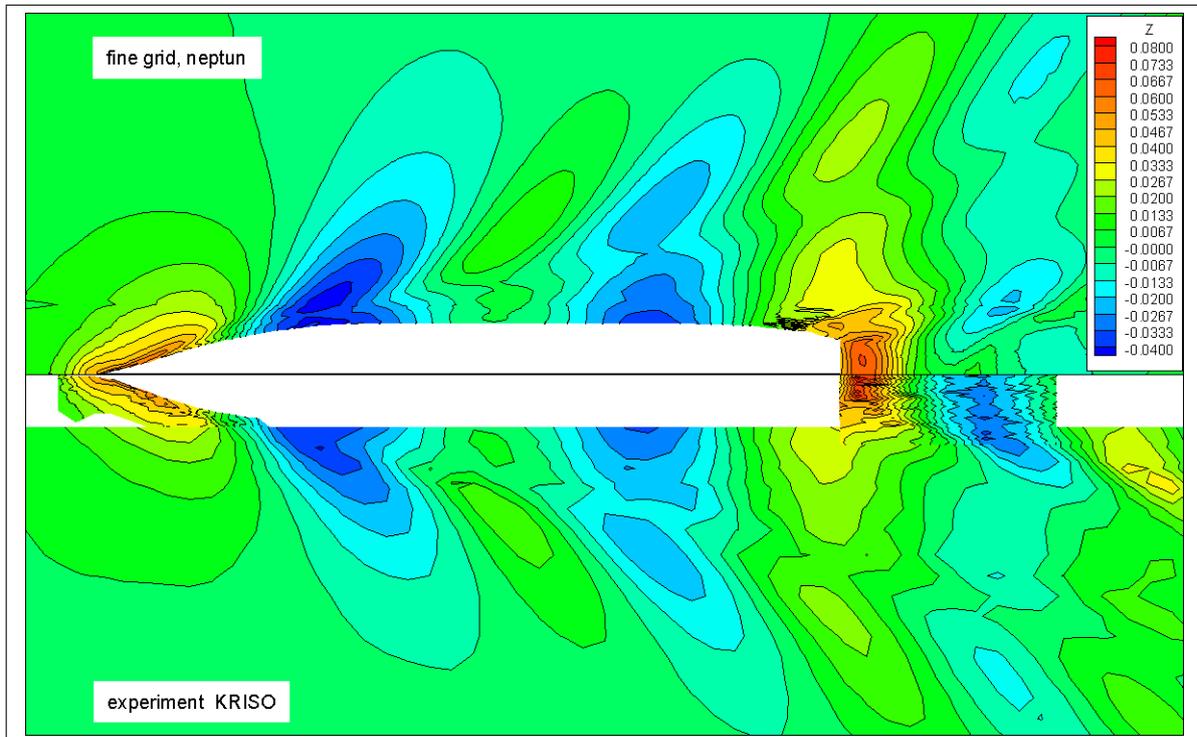


Abbildung 11 Wellenfeld des KCS mit Neptun-Software (Gothenburg 2000)  
*Quelle: Archivdaten von Prof. Dr. Andrés Cura*

## 8 Abschlussbetrachtung

Im Laufe dieser Arbeit wurde das KRISO Container Ship unter Glattwasserbedingungen ausführlich per Simulation untersucht. Die erreichten Ergebnisse sind hauptsächlich akademisch relevant. Sie zeigen die Möglichkeiten und Grenzen von OpenFOAM auf. Gleichzeitig kann man die erzielten Ergebnisse auch als Vorarbeit für weitere Recherchen und Experimente auf diesem Gebiet sehen.

Bei Betrachtung der Validierungsergebnisse stellt man fest, dass es weiterer Recherche bedarf, um alltagstaugliche Ergebnisse für (zum Beispiel) eine Werft zu produzieren. Es summieren sich verschiedene Faktoren, die die Ergebnisse von den vorliegenden Experimenten abweichen lassen.

Zum einen berechnet OpenFOAM einen überzogenen Reibungswiderstand. Das kann möglicherweise an der Gitterqualität liegen. Die Probleme von `snappyHexMesh` wurden ja bereits beschrieben. Gleichzeitig ermöglicht dieser automatisierte Algorithmus zur Gittererstellung aber auch eine große zeitliche Ersparnis im Vergleich zu anderen visuellen Gittererstellungstools. Diese grobe Gittererzeugung kann für schnell zu ermittelnde erste Erkenntnisse in neuen Bereichen durchaus sinnvoll sein. Bei genaueren Analysen sollte man anschließend jedoch in bessere Gitter investieren.

Zum anderen stellt die reflektierende Welle in der Simulation mit freier Oberfläche eine erhebliche Störung dar. Andere OpenFOAM Nutzer haben diese Erfahrung auch schon gemacht, bisher fand aber niemand eine zufriedenstellende Lösung.

Die Lösungsalgorithmen (Solver) und -programme in OpenFOAM sind sehr gut. Sie verfügen über vielfältige Einstellmöglichkeiten und bei Bedarf kann man sogar umprogrammieren. Ein kleines Hindernis bei der Nutzung von OpenFOAM besteht allerdings darin, dass die verfügbare Dokumentation sehr knapp gehalten ist. Das offizielle Handbuch zur Software gibt nur eine kleine Anzahl an Einstellmöglichkeiten wieder und ist nur in geringem Maße hilfreich. Bei tiefgreifenden Fragen kann es nützlich sein, das Online-Forum "CFD-online" <http://www.cfd-online.com/Forums/openfoam/> zu besuchen. Die Autoren der Foreneinträge haben mir mit ihren Fragen und Antworten oft geholfen oder wichtige Ideen geliefert.

Einer der wichtigsten Bestandteile dieser Arbeit ist die Verifikation und Validierung (engl. Uncertainty Analysis, UA) der Ergebnisse. Diese kritische Analyse der Simulationsergebnisse stellt in meinen Augen den Schlüssel zur globalen Anerkennung von CFD-Rechnungen dar. Es könnte als Ziel der numerischen Strömungsmechanik im Bereich Schiffstechnik angesehen werden, Modellversuche komplett durch Simulationen zu ersetzen. So etwas kann mit einer Kombination ausführlicher UA und zusätzlichem Vergleich von Simulationen verschiedener Codes, in mittelfristiger Zukunft möglich werden. Der Weg wird auf jeden Fall über eine standardisierte und ausführliche Validierung der Ergebnisse führen.

## 9 Zusammenfassung

Das KCS wurde mit der CFD Software OpenFOAM ausführlich untersucht. Von den berechneten Variablen in der Simulation, wurden sechs Stück einer ausführlichen Verifikation und Validierung unterzogen. Dabei konnten zwei Variablen validiert werden. Der korrigierte Gesamtwiderstandsbeiwert ( $C_{TC} = 3,958 \cdot 10^{-3}$ ) wurde mit  $11,86\%C_T$  Unsicherheit validiert, wie auch größtenteils die normierte Nachstromgeschwindigkeit  $V_1$ . Die restlichen Ergebnisse konnten nicht validiert werden, da meistens der Konvergenzradius  $R_k$  divergierte oder stark in den negativen Bereich fiel. Um weitere Erkenntnisse zu gewinnen, müsste eine weitere Gitterstudie mit einem nochmals feinerem Gitter (Zellenanzahl des feinen Gitters  $\cdot 1,5^3$ , also mit ca. 2,4 Millionen Zellen) durchgeführt werden, was die Möglichkeit für eine weiterführende Arbeit bietet. Es könnten bei einer zweiten Gitterstudie möglicherweise mehr Variablen validiert werden.

Da es insgesamt für den Fall der monotonen Konvergenz des  $R_k$  nötig ist, dass die Variation der Lösung ( $\varepsilon_{k_{ij}}$ ) zwischen den gröberen Gittern größer ist als zwischen den feinen, könnte der Gesamtwiderstandsbeiwert höchstens auf  $C_T = 3,91 \cdot 10^{-3}$  gesenkt werden. Das entspricht immer noch einer erheblichen Überschätzung der Widerstandskräfte. Somit würde ein feineres Gitter die Überschätzung des Reibungswiderstandes nicht eliminieren, sondern nur verkleinern.

Alle der Validierung unterzogenen Ergebnisse weichen ca. 10% vom Experiment ab. Ein feineres Gitter könnte erwartungsgemäß genauere Ergebnisse liefern. Vor allem aber sollte das Gitter mit mehr Widmung zum Detail erstellt werden. Die Vermutung liegt nahe, dass `snappyHexMesh` durch seinen Automatismus ein Gitter erstellt, welches unsaubere Bereiche vorweist. Dadurch ergibt sich eine kompliziertere und fehleranfälligere Aufgabe für das Lösungsprogramm, was in OpenFOAM in einem zu hohen Reibungswiderstand und somit zu hohen Gesamtwiderstand führt.

Der Nachstrom und das Wellenbild des KCS werden in den Simulationsergebnissen zufriedenstellend wiedergegeben. Im direkten Vergleich mit den experimentellen Daten, sowie mit den Ergebnissen der Gothenburg 2000 CFD-Workshopteilnehmer, sind die Ergebnisse der beiden Felder positiv zu werten. Negativ dagegen fällt die starke Abweichung des Formfaktors auf. OpenFOAM berechnet diesen auf dem feinen Gitter zu 22,11%, was einer Abweichung von über 100% gegenüber dem erwarteten Wert ( $k = 10\%$ ) aus dem Experiment entspricht. Das liegt am großen Druckanteil im viskosen Widerstandsbeiwert  $C_V$ , der wahrscheinlich durch numerische Diffusion zustande kommt.

Entscheidend fällt bei der OpenFOAM Simulation mit freier Oberfläche die lang anhaltende, reflektierende Welle auf. Sie bringt erhebliche iterative Unsicherheiten in den Validierungsprozess ein und sorgt für eine periodische Schwankung der gemessenen Druckkraft. Dieses Problem muss auf jeden Fall weiter untersucht werden. Im CFD-online Forum [6] wurde die Möglichkeit diskutiert, eine Dämpfungsfunktion in `interFOAM` zu integrieren, welche Wellen kurz vor dem Auslass eliminiert. Dies wäre ein großer Aufwand, der viel

Programmierkenntnis erfordert und ein potentielles Thema für eine weiterführende Arbeit darstellt.

Insgesamt wäre es sinnvoll eine Wissensbasis für Schiffsberechnungen mit OpenFOAM aufzubauen. Diese Basis sollte Anleitungen, Probleme und implementierte Lösungen enthalten, um einen schnellen Einstieg in dieses Softwarepaket zu ermöglichen. Denn Open Source CFD-Software wie OpenFOAM, wird in Zukunft mehr und mehr gefragt sein, da ihr Quellcode beliebig erweiterbar ist. Durch eben die Eigenschaft des zugänglichen Quellcodes von OpenFOAM, lassen sich neue Erweiterungen und Implementierungen realisieren. Diese müssen natürlich anschließend ausreichend dokumentiert werden.

OpenFOAM wird ohnehin fortlaufend von der Firma OpenCFD weiterentwickelt und verbessert. Zum Abgabetermin dieser Arbeit ist bereits die Version OpenFOAM 1.7.1 veröffentlicht worden. Es wäre sehr schade dieses große, sich für den Schiffbau anbietende Potenzial, nicht in weiteren Forschungsarbeiten zu nutzen.

## Literatur

- [1] Gothenburg 2000, workshop on ship hydrodynamics, kcs data. Website, 06. Aug. <http://www.nmri.go.jp/cfd/cfdws05/gothenburg2000/KCS/container.html>.
- [2] *ITTC Recommended Procedures and Guidelines; Uncertainty Analysis in CFD; Verification and Validation*, 2002.
- [3] *OpenFOAM user guide ver. 1.6*, 24. July 2009.
- [4] M. Cura Hochbaum, A. & Vogt. Flow and resistance prediction for a container ship. In *Gothenburg 2000, A Workshop on Numerical Ship Hydrodynamics*, 2002.
- [5] Peric M. Ferziger, J.H. *Numerische Strömungsmechanik*. Springer Verlag, 2008.
- [6] Forum. Online center for computational fluid dynamics. Website, Okt. 2010. <http://www.cfd-online.com>.
- [7] et al. Ken-ichi Kume. Measurements of surface pressure and local velocity field around a kcs model and uncertainty analysis. Technical report, Ship Research Institute, Tokyo, Japan, 2000.
- [8] F; Bertram V Larsson, L; Stern, editor. *Gothenburg 2000, A Workshop on Numerical Ship Hydrodynamics*, number CHA/NAV/R-02/0073. Chalmers University of Technology, 2002.
- [9] Oertel H. jr Laurien, E. *Numerische Strömungsmechanik*. vieweg+Teubner, 3 edition, 2009.
- [10] OpenCFD. Openfoam website, Okt 2010. <http://www.openfoam.com>.
- [11] Thamsen Siegmann, H.E. *Strömungslehre*. Springer Verlag, 2008.
- [12] sname. Ittc recommended procedures. website, Okt. 2010. <http://ittc.sname.org/documents.htm>.
- [13] Wilson R. V. Coleman H. W. Paterson E. G. Stern, F. Verification and validation of cfd simulations. Technical Report IIHR No. 407, Iowa Institute of Hydraulic Research, 1999.
- [14] G. Tampier. Mesh generation for ship hydrodynamics, problems with snappyhexmesh. Technical report, Technische Universität Berlin, 2010.

# Anhang

## A Daten des KRISO Containerschiffs

Bezeichnung	Wert
Länge zwischen den Loten	$L_{PP} = 7,2786m$
Breite	$B = 1,019m$
Tiefgang	$T = 0,342m$
Benetzte Fläche	$S_0 = 9,4379m^2$
Blockkoeffizient	$C_B = 0,65$
Reynolds-Zahl	$R_n = 1,4 \cdot 10^7$
Froude-Zahl	$F_n = 0,26$
Erdbeschleunigung	$g = 9,8062 \frac{m}{s^2}$

**Tabelle 12 Hauptabmessungen KCS**

Modellgeschwindigkeit für die gegebene Froude-Zahl:

$$F_n = \frac{U_M}{\sqrt{g \cdot L_{PP}}} \iff U_M = F_n \cdot \sqrt{g \cdot L_{PP}}$$

$$U_M = 0,26 \cdot \sqrt{9,8062 \frac{m}{s^2} \cdot 7,2786m} = 2,1966 \frac{m}{s}$$

### Experimentell ermittelte Daten[1] für Validierung

Bezeichnung	Wert	Unsicherheit $U_D$
Gesamtwiderstandsbeiwert	$C_T = 3,56 \cdot 10^{-3}$	$U_{D_{CT}} = 0,64\%$
Restwiderstandsbeiwert	$C_R = 0,731 \cdot 10^{-3}$	$U_{D_{CR}} = 0,64\%$
Reibungswiderstandsbeiwert	$C_{F_{ITTC}} = 2,832 \cdot 10^{-3}$	$U_{D_{CR}} = 0,64\%$
Formfaktor	$k = 10\% *$	$U_{D_k} = 5\%$
Wellenlängsschnitt	siehe Abb. 10	$U_{D_{WC}} = 2,4\%$
axialer Geschwindigkeitsschnitt	siehe Abb. 7 (Symbolverlauf)	$U_{D_{VC}} = 0,38\%$

**Tabelle 13 Experimentelle Daten**

\* Der Wert für  $k$  wurde aus einem Paper über die KCS Experimente für den Gothenburg 2000 Workshop entnommen [7].

## B Gitterparameter

### Randbedingungen für Lösungsgenerierung ohne FO

Boundary conditions

<b>Grenze</b>	<b>k</b>	<b>omega</b>
minX	fest $k = 7,12 \cdot 10^{-4}$	fest $\omega = 712$
maxX	Null Gradient	Null Gradient
minZ	Null Gradient	Null Gradient
maxZsym	Symmetrieebene	Symmetrieebene
minY	Symmetrieebene	Symmetrieebene
maxY	Null Gradient	Null Gradient
ship	k-Wandfunktion	omega-Wandfunktion
Volumen	$k = 7,12 \cdot 10^{-4}$	$\omega = 712$
<b>Grenze</b>	<b>p</b>	<b>U</b>
minX	fest $p = 0$	fest $U = (2,1966 \ 0 \ 0)$
maxX	Null Gradient	Null Gradient
minZ	Null Gradient	slip
maxZsym	Symmetrieebene	Symmetrieebene
minY	Symmetrieebene	Symmetrieebene
maxY	Null Gradient	slip
ship	Null Gradient	fest $U = (0 \ 0 \ 0)$
Volumen	$p = 0$	$U = (2,1966 \ 0 \ 0)$

Tabelle 14 Randbedingungen der Simulation ohne FO

## Randbedingungen für Lösungsgenerierung mit FO

Boundary conditions

Grenze	alpha1	k	omega
minX	groovyBC	fest $k = 7,12 \cdot 10^{-4}$	fest $\omega = 712$
maxX	Null Gradient	Null Gradient	Null Gradient
minZ	Null Gradient	Null Gradient	Null Gradient
maxZ	Einlass & Auslass	Einlass & Auslass	Einlass & Auslass
minY	Symmetrieebene	Symmetrieebene	Symmetrieebene
maxY	slip	Null Gradient	Null Gradient
ship	Null Gradient	k-Wandfunktion	omega-Wandfunkt.
Volumen	definiertes Feld	$k = 7,12 \cdot 10^{-4}$	$\omega = 712$
Grenze	p	U	
minX	Null Gradient	fest $U = (2,1966 \ 0 \ 0)$	
maxX	hydrostat. Druck	Null Gradient	
minZ	hydrostat. Druck	slip	
maxZ	Druckeinlass	geschw./Druckauslass	
minY	Symmetrieebene	Symmetrieebene	
maxY	hydrostat. Druck	slip	
ship	hydrostat. Druck	fest $U = (0 \ 0 \ 0)$	
Volumen	$p = 0$	$U = (2,1966 \ 0 \ 0)$	

Tabelle 15 Randbedingungen der Simulation mit FO

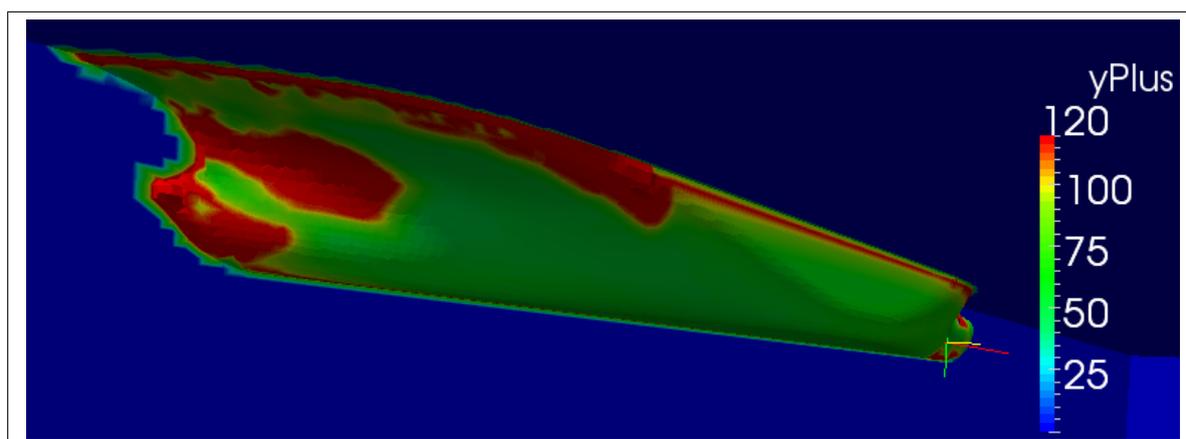


Abbildung 12 yPlus des feinen Gitters ohne FO

## C Rechnungen und Graphen des Validierungsverfahrens

### Tabellarische Übersicht der Validierungsergebnisse

Variable	S	D	E %	$U_V$ %	validiert
$C_V$	$3,29 \cdot 10^{-3}$	$2,832 \cdot 10^{-3}$	16,2% <i>D</i>	3,04% <i>D</i>	nein
$k$	22,11 %	22,11 %	-	-	nein
$V1$	Tab. 17	Tab. 17	Tab. 18	Tab. 18	größtenteils
$C_T$	$4,187 \cdot 10^{-3}$	$3,56 \cdot 10^{-3}$	17,61% <i>D</i>	14,9% <i>D</i>	nein
$C_{TC}$	$3,958 \cdot 10^{-3}$	$3,56 \cdot 10^{-3}$	11,19% <i>D</i>	11,86% <i>D</i>	ja
$C_R$	$0,535 \cdot 10^{-3}$	$0,731 \cdot 10^{-3}$	-	-	nein
$\frac{z}{L}$	Abb. 10	Abb. 10	-	-	nein

Tabelle 16 Zusammenfassung der Validierungsergebnisse

### C.1 Gitter ohne FO

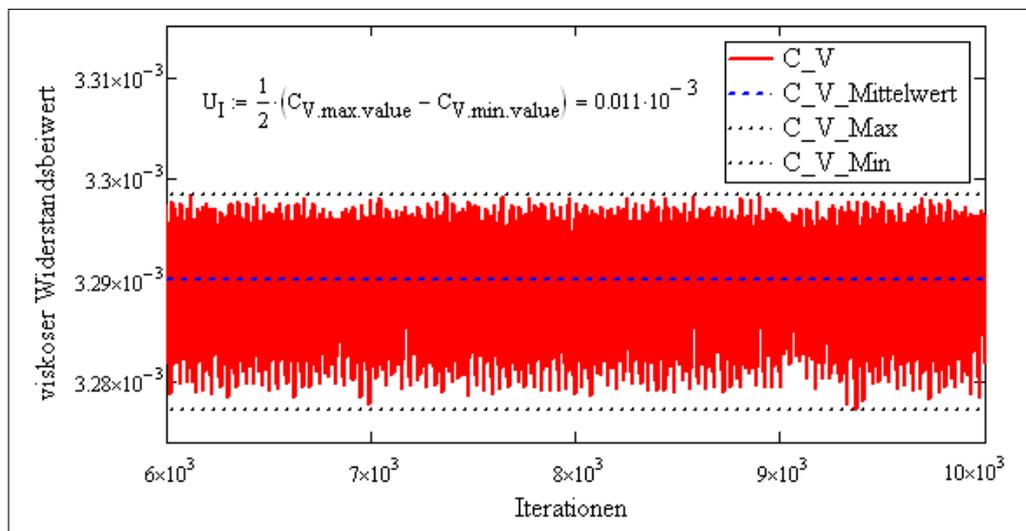


Abbildung 13 Iterative Unsicherheit von  $C_V$  beim Gitter ohne FO

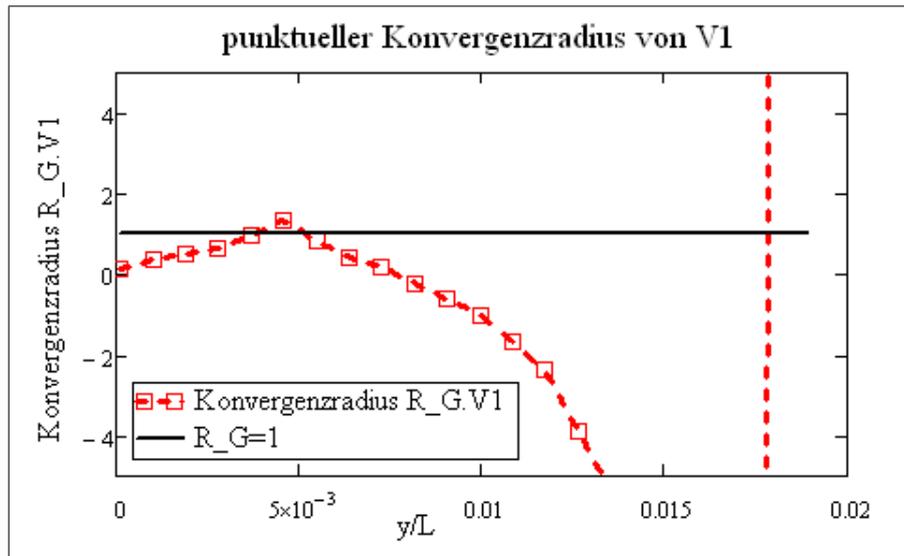


Abbildung 14 Konvergenzradius von V1 in jedem Punkt

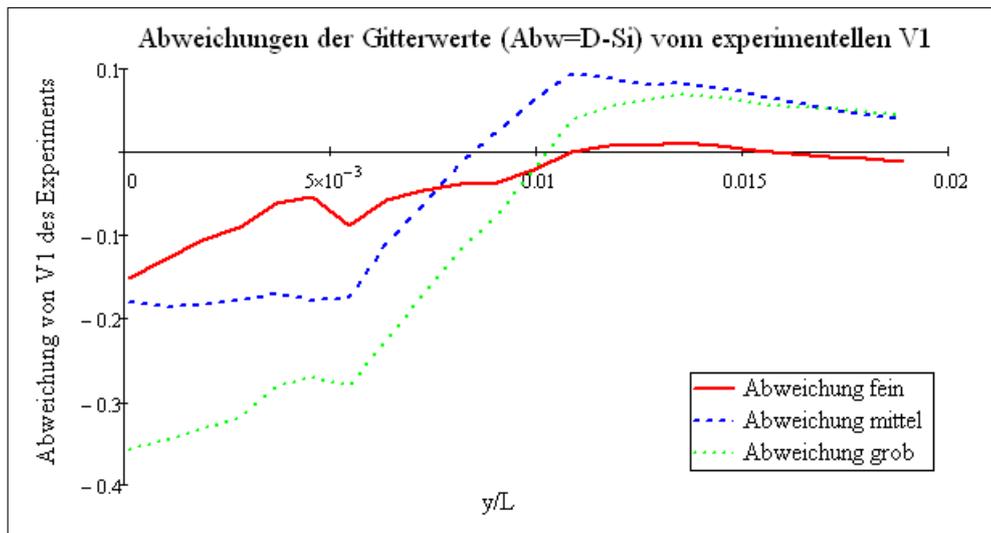


Abbildung 15 Abweichung des berechneten V1 vom Experiment

## Werteübersicht der Validierung von V1

Nr	y/L	$V1_{Exp}$	$V1_{fein}$	$V1_{mittel}$	$V1_{grob}$	$R_{GV1}$	$\delta_{REG}^*$
1	0,00012	0,2864	0,439	0,467	0,645	0,157	0,0969
2	0,00101	0,2926	0,421	0,478	0,638	0,363	0,2007
3	0,00190	0,2989	0,405	0,481	0,631	0,506	0,2639
4	0,00280	0,3039	0,394	0,482	0,623	0,626	0,3048
5	0,00369	0,3320	0,393	0,502	0,613	0,970	0,3753
6	0,00459	0,3601	0,414	0,538	0,630	1,355	0,4318
7	0,00548	0,3889	0,476	0,565	0,670	0,836	0,3057
8	0,00638	0,4817	0,539	0,591	0,710	0,438	0,1802
9	0,00727	0,5757	0,622	0,639	0,747	0,163	0,0609
10	0,00817	0,6670	0,706	0,681	0,784	-0,241	-0,0858
11	0,00906	0,7313	0,768	0,707	0,806	-0,617	-0,2131
12	0,00995	0,7951	0,816	0,733	0,815	-1,008	-0,2899
13	0,01085	0,8591	0,859	0,765	0,821	-1,668	-0,3257
14	0,01174	0,8813	0,873	0,793	0,827	-2,376	-0,2764
15	0,01264	0,8943	0,887	0,813	0,832	-3,893	-0,2562
16	0,01353	0,9073	0,895	0,825	0,838	-5,345	-0,2448
17	0,01443	0,9097	0,901	0,833	0,844	-6,062	-0,2373
18	0,01532	0,9093	0,907	0,841	0,851	-6,696	-0,2280
19	0,01622	0,9088	0,911	0,849	0,855	-10,324	-0,2142
20	0,01711	0,9096	0,915	0,857	0,858	-97,191	-0,2002
21	0,01801	0,9101	0,918	0,865	0,863	22,616	-0,1834
22	0,01890	0,9106	0,921	0,871	0,867	13,598	-0,1751

Tabelle 17 Auswertung von V1 - Teil 1

### Bemerkungen zu Tabelle 17

- Nr ist die Nummerierung der Punkte
- y/L ist die normierte y-Koordinate des Punktes
- $V1_{Exp}$  ist das Ergebnis der Experimentellen Versuche.  $V1_{Gittertyp}$  gibt jeweils den interpolierten Wert des feinen, mittleren und groben Gitters an.
- $R_{GV1}$  ist der Konvergenzradius in jedem Punkt
- $\delta_{REG}^*$  ist der ermittelte Fehler des feinen Gitters per RE

Nr	$U_{G.C_{V1}}$	$\delta_{G_{V1}}^*$	$U_{G_{V1}}$	$S_{C_{V1}}$	$E_{V1}$	$E_{V1C}$	$U_{V_{V1}}$	$U_{VC_{V1}}$
1	0,0745	0,02238	0,0969	0,4167	-0,1526	-0,1302	0,097	0,074
2	0,1543	0,04636	0,2007	0,3742	-0,1279	-0,0815	0,201	0,154
3	0,2029	0,06096	0,2639	0,3436	-0,1057	-0,0447	0,264	0,203
4	0,2344	0,07041	0,3048	0,3240	-0,0906	-0,0202	0,305	0,234
5	0,2886	0,08669	0,3753	0,3067	-0,0614	0,0253	0,375	0,289
6	0,3320	0,09974	0,4318	0,3142	-0,0537	0,0460	0,432	0,332
7	0,2351	0,07062	0,3057	0,4059	-0,0876	-0,0170	0,306	0,235
8	0,1386	0,04163	0,1802	0,4973	-0,0572	-0,0156	0,180	0,139
9	0,0469	0,01408	0,0609	0,6076	-0,0460	-0,0319	0,061	0,047
10	0,0660	-0,01983	0,0858	0,7260	-0,0392	-0,0590	0,086	0,066
11	0,1639	-0,04924	0,2131	0,8175	-0,0369	-0,0861	0,213	0,164
12	0,2230	-0,06698	0,2899	0,8831	-0,0210	-0,0880	0,290	0,223
13	0,2505	-0,07525	0,3257	0,9340	0,0004	-0,0749	0,326	0,251
14	0,2125	-0,06385	0,2764	0,9367	0,0084	-0,0555	0,276	0,213
15	0,1970	-0,05918	0,2562	0,9465	0,0070	-0,0522	0,256	0,197
16	0,1882	-0,05655	0,2448	0,9520	0,0119	-0,0446	0,245	0,188
17	0,1825	-0,05481	0,2373	0,9562	0,0083	-0,0465	0,237	0,183
18	0,1753	-0,05266	0,2280	0,9593	0,0026	-0,0500	0,228	0,175
19	0,1647	-0,04947	0,2142	0,9603	-0,0020	-0,0515	0,214	0,165
20	0,1540	-0,04626	0,2002	0,9612	-0,0054	-0,0517	0,200	0,154
21	0,1411	-0,04237	0,1834	0,9604	-0,0080	-0,0504	0,183	0,141
22	0,1347	-0,04046	0,1751	0,9617	-0,0107	-0,0512	0,175	0,135

**Tabelle 18** Auswertung von V1 - Teil 2

### Bemerkungen zu Tabelle 18

- $U_{G.C_{V1}}$  ist die korrigierte Unsicherheit für  $C_{G_{V1}}$  nahe 1
- $\delta_{G_{V1}}^*$  korrigierter abgeschätzter Fehler aufgrund des Gitters
- $U_{G_{V1}}$  ist die Unsicherheit für  $C_{G_{V1}}$  kleiner 1
- $S_{C_{V1}}$  ist das korrigierte Ergebnis des feinen Gitters  $S_{C_{V1}} = V1_{fine} - \delta_{G_{V1}conf}$
- $E_{V1}$  und  $E_{V1C}$  ist jeweils der Vergleichsfehler und der korrigierte Vergleichsfehler
- $U_{V_{V1}}$  und  $U_{VC_{V1}}$  ist jeweils die Validierungsunsicherheit und die korrigierte Validierungsunsicherheit

## Validierungsunsicherheit von V1

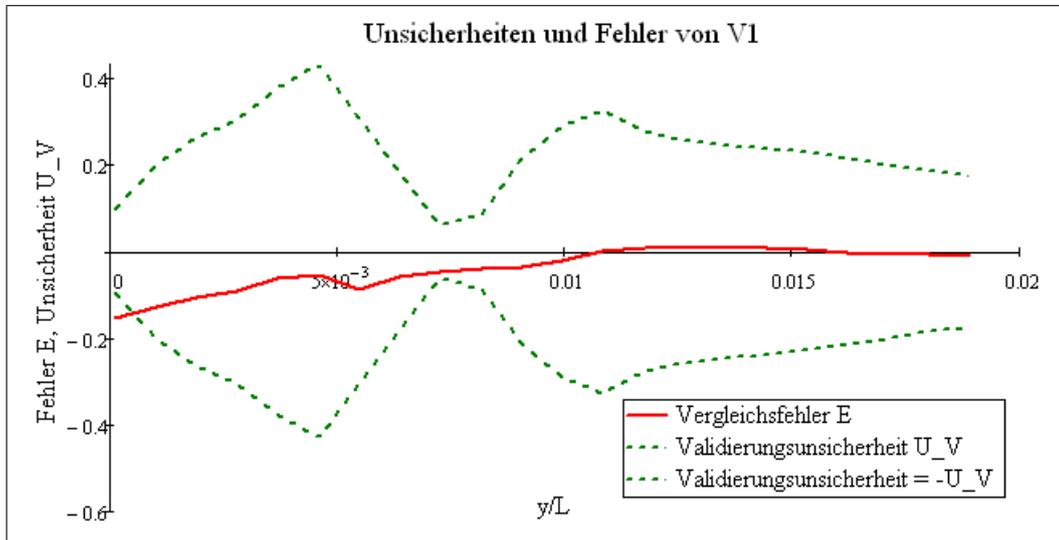


Abbildung 16 Vergleich - Unsicherheiten und Fehler von V1

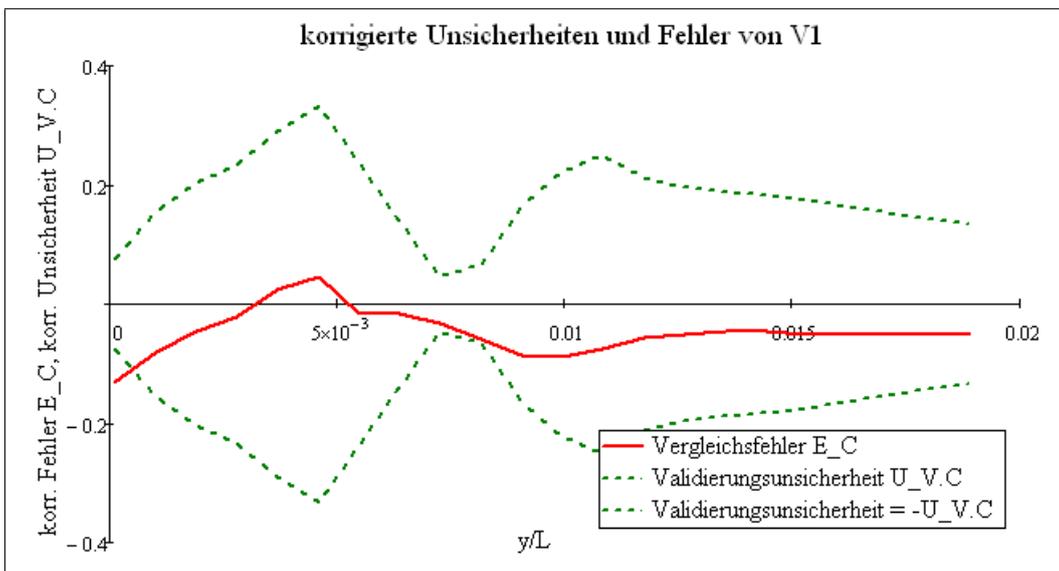


Abbildung 17 Vergleich - korrigierte Unsicherheiten und Fehler von V1

## C.2 Gitter mit FO

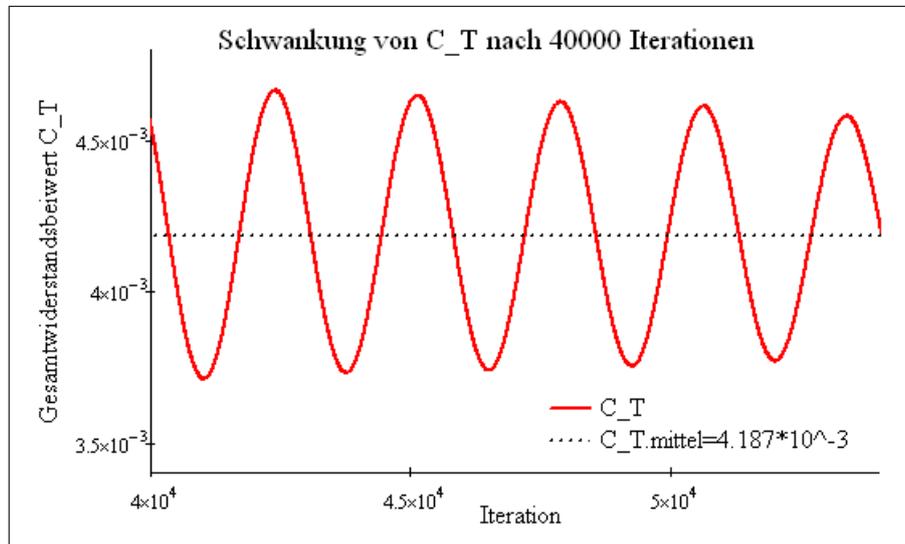


Abbildung 18 Schwankung von  $C_T$  des feinen Gitters nach 40000 Iterationen

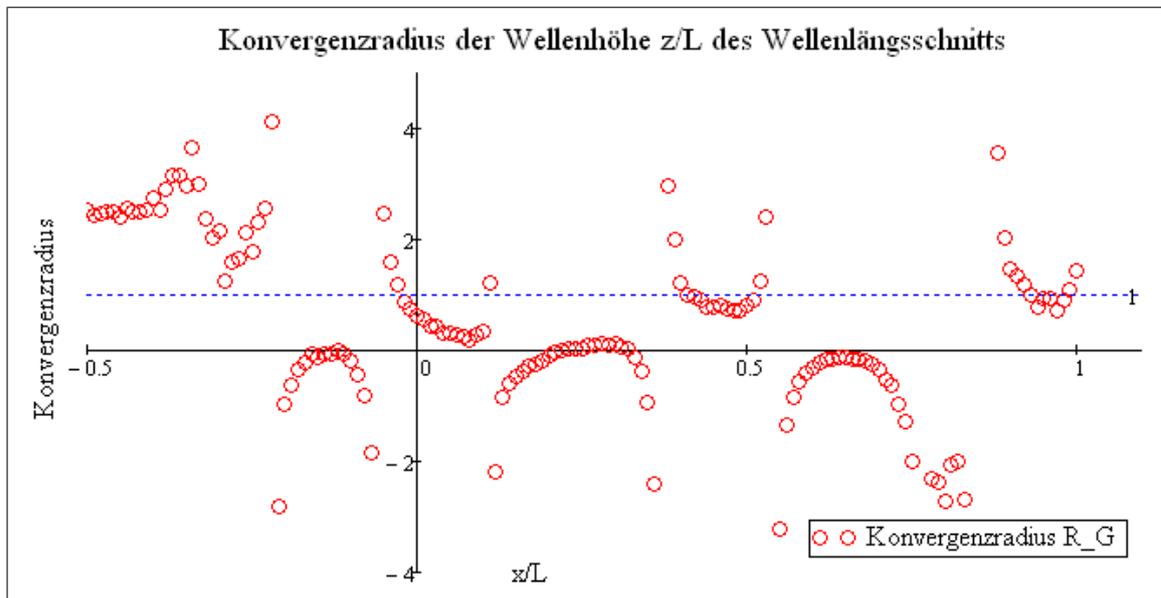


Abbildung 19 punktueller Konvergenzradius der Wellenhöhe  $z/L$  im Wellenlängsschnitt ( $y/L=0,1509$ )

### C.3 Ergebnisse von Gothenburg 2000

Institution	Code	$C_T * 10^3$	$C_F * 10^3$	$C_P * 10^3$	$ E  \%$	$U_V \%$	$C_T$
-	Exp.	3,56	-	-	-	$U_D = 1\%$	
HUT-VVT	FINFLO	3,57	2,89	0,68	0,28	4,1	
SRI	SURF	3,92	3,13	0,79	10,11	1,13	
HSVA-ICCM	COMET	3,94	2,58	1,36	10,67	-	
HSVA	NEPTUN	3,93	2,96	0,97	10,39	-	
MARIN-IST	PARNASSOS	3,50	3,13	0,37	1,68	-	
OPU	FLOWPACK	3,59	2,93	0,66	0,84	-	
SVA-AEA	CFX	3,65	2,72	0,93	2,53	-	
USDDC	UVW	3,78	2,75	1,03	6,18	-	
-	Mittel	3,74	2,89	0,85	5,34	2,62	
-	Std. Abw.	0,19	0,20	0,30	-	-	
-	Abw. [%]	5,10	6,90	35,3	-	-	

Tabelle 19 Gothenburg 2000, Ergebnisse der Teilnehmer

## D Genutzte Funktionen von Linux und OpenFOAM

### Häufig genutzte Linux-Anwendungen

- **tail**: zeigt die letzten Zeilen einer Textdatei an. Um beispielsweise die letzten 200 Zeilen einer Datei zu sehen ist folgende Schreibweise nötig: `tail -n 200 dateiName`
- **cat**: zeigt den Inhalt einer Datei an und springt danach in die Kommandozeile zurück.
- **vi**: ein nützlicher Allzweckeditor unter Linux

### forcesInter Eintrag im controlDict

Folgende Zeilen wurden dem *controlDict* am Dateiende angefügt um die Ausgabe der Kräfte zu erhalten.

```
functions
(
  forcesInter
  {
    type forcesInter;                // new class!
```

```

        functionObjectLibs ("libforces.so");           // req. lib.
        patches (ship);                               // name of patches
        pName p;
        Uname U;
        rhoInf 1000.0;                                // ref. density
        rhoName rhoInf;
        CofR (0. 0 0); // origin center of rotationlibforces.dylib
        outputControl    timeStep;
        outputInterval  3;
    }
);

```

## Beispielscript zur Gittererstellung (FO)

```

#!/bin/sh

# shrinking factor
DESCALE=0.625
# 1/shrinking factor
RESCALE=$(echo "scale=5;1/$DESCALE" | bc)
SHIPDIR=shipbox
# blockMeshDictionary
BMDIC=constant/polyMesh/blockMeshDict

#####
# OUTER BOX
rm -rf 1 2 3 4 5 6 7 8 9 0.*
# create BMDIC from m4 definitions for the outer box
m4 $BMDIC.m4 > $BMDIC
blockMesh
# create a cell set dictionary for wave zone refinement from m4
m4 system/cellSetDict_waves.m4 > system/cellSetDict
# create cell set
cellSet
# prepare refineMesh dictionary
cp system/refineMeshDict_waves system/refineMeshDict
# refine mesh in wave zone
refineMesh -dict
# create cell set dictionary for free surface zone re-refinement from m4
m4 system/cellSetDict_fsurf.m4 > system/cellSetDict
cellSet
cp system/refineMeshDict_fsurf system/refineMeshDict

```

```

# re-refine mesh in free surface zone
refineMesh -dict

#####
# SHIP BOX
# create BMDIC from m4 definitions for the ship box
m4 $$SHIPDIR/$BMDIC.m4 > $$SHIPDIR/$BMDIC
# create snappyDict from m4 definitions
m4 $$SHIPDIR/system/snappyHexMeshDict.m4> $$SHIPDIR/system/snappyHexMeshDict
cd $$SHIPDIR
rm -rf 1 2 3 4 5 6 7 8 9 0.*
rm log.*
./scale.awk $DESCALE
blockMesh
transformPoints -scale "($DESCALE 1 1)"
snappyHexMesh | tee log.snappyHexMesh
transformPoints -scale "($RESCALE 1 1)"
# create patch unifying "hull" and "deck" into "ship" patch
createPatch
checkMesh -latestTime | tee log.checkMesh
# foamToVTK
rm -rf 1 2
cd ..

#####
# STITCHING
mergeMeshes .. ${PWD##*/} . $$SHIPDIR
stitchMesh boxwater boxship
# symmetryPlane patch unifying symmetryPlanes - outer box and ship box
createPatch
foamToVTK #-latestTime
checkMesh -latestTime | tee log.checkMesh
rm -rf 1 2 3 4

```