

Februar 2019

SCHRIFTENREIHE SCHIFFBAU

Dennis Mierke, Martin Gehrke,
Christian F. Janßen, Thomas Rung

MOUSE

Modellierung der Umströmung von Schiffen in Eisbedeckten Gewässern

Schlussbericht zum Forschungsvorhaben im Verbundprojekt ProEis
(Forschungskennziffer 03SX391G)

TUHH

Technische Universität Hamburg

BMW*i* Verbundforschungsvorhaben ProEis
Einfluss der Formgebung von Schiffen auf die Propulsionseffizienz und die
Propeller-Eis-Belastung von Schiffen



Schlussbericht des Teilvorhabens

MOUSE

Modellierung der Umströmung von Schiffen in Eisbedeckten Gewässern

M.Sc. D. Mierke, M.Sc. M. Gehrke, Dr.-Ing. C.F. Janßen, Prof. Dr.-Ing. T. Rung

Februar 2019

Forschungskennziffer: **03SX391G**

Verbundprojekt: *ProEis* –
Einfluss der Formgebung von Schiffen auf die Propulsionseffizienz und die Propeller-Eis-Belastung von Schiffen

Verbundpartner: Hamburgische Schiffbau-Versuchsanstalt GmbH (HSVA),
DNV GL SE (DNVGL), Mecklenburger Metallguss GmbH
(MMG), Voith GmbH & Co. KGaA (Voith), MV WERF-
TEN Wismar GmbH (MVW) [ehemals Nordic Yards Wis-
mar GmbH], develogic GmbH (dev), Technische Universität
Hamburg (TUHH)

Laufzeit: 01.07.2015 bis 30.06.2018 (30.09.2019)

Teilprojekt: *MOUSE* –
Modellierung der Umströmung von Schiffen in Eisbedeckten
Gewässern

Forschungsstelle: Technische Universität Hamburg
Institut für Fluidodynamik und Schiffstheorie (M-8)
Am Schwarzenberg-Campus 4, D-21073 Hamburg

Autoren: M.Sc. D. Mierke, M.Sc. M. Gehrke,
Dr.-Ing. C.F. Janßen, Prof. Dr.-Ing. T. Rung

Inhaltsverzeichnis

I	Kurzdarstellung des Teilvorhabens	1
1	Zielsetzung und Motivation	1
2	Ausgangssituation	4
2.1	Entwurfsmethoden auf der Basis viskoser Hydrodynamik	4
2.2	Entwurfsmethoden für den Verkehr in eisbedeckten Gewässer	4
2.3	Propulsion in Eis und Propeller-Eis-Interaktion	6
3	Planung und Ablauf	7
4	Stand der Forschung vor Projektbeginn	9
5	Zusammenarbeit mit anderen Forschungsstellen	10
II	Eingehende Darstellung des Teilvorhabens	11
1	Einleitung	11
2	Numerische Simulation mit dem Gitter-Boltzmann-Verfahren	12
2.1	Grundgleichungen des Gitter-Boltzmann-Verfahrens	12
2.2	Implementierung	14
2.3	MRT-Kollisionsoperator	15
2.4	Grobstruktursimulation – Smagorinsky Turbulenzmodell	17
2.5	Freie-Oberflächen-Modell	18
2.5.1	VoF-Erweiterungen an <code>elbe</code>	20
2.6	Gitterverfeinerung – Zonale Methoden des Fluidlösers	21
2.6.1	LBM – Verteilungsfunktionen	22
2.6.2	VOF – Füllstandsvariable	24
3	Physik-Bibliotheken	29
3.1	Verfügbare Bibliotheken	30
4	Open Dynamics Engine	32
4.1	Kollisionsmethode	33
4.1.1	Kollisionserkennung	34
4.1.2	Kollisionsbehandlung	36
4.2	Lösungsverfahren	38
4.3	Implementierung	39
4.4	Validierung	41
4.4.1	Kollisionsbehandlung	41
4.4.2	Reibungsmodelle	43
4.4.3	Kollisionsaufwand	47
4.4.4	Zusammenfassung	48

5	Bidirektionale Fluid-Struktur-Interaktion	50
5.1	Kraftberechnungsverfahren	50
5.1.1	Impulsaustauschmethode über die Gitterknoten	51
5.1.2	Spannungsintegrationsmethode über die Gitterknoten	51
5.1.3	Spannungsintegrationsmethode über die Geometrie	52
6	Numerische Methoden zur Effizienzsteigerung	54
6.1	Indirekte Adressierung des freie-Oberflächen-Modells	54
6.2	Hüllkörper-basierte Haftrandbedingung und Kraftberechnung	54
6.3	Parallelisierte Addition	56
6.4	stream-Parallelisierung	57
7	Vereinfachtes Propulsionsmodell	59
8	Eisschollengenerator	61
9	Messmethode zur Analyse des Eisdurchflusses in der Propellerebene	64
10	Algorithmus zur Neuvergitterung aufbrechender Geometrien	66
11	Haftrandbedingung höherer Ordnung	69
12	Anpassungen an die Simulation eines Voith-Schneider-Propellers (VSP)	70
13	Anwendung und Validierung des entwickelten numerischen Verfahrens	74
13.1	Simulation des Eis-Volumenstroms am mo4977-Schiffsmodell (Azi-Pod Testfall)	75
13.2	Voith-Schneider-Propeller Testfall	80
13.2.1	Numerische Umsetzung des Testfalles	81
13.2.2	VSP Validierung und Parameterstudie für Simulationen eines generischen Rumpfes im Modellmaßstab	86
13.2.3	VSP Validierung unter Freifahrtbedingungen	88
13.2.4	VSP Validierung unter Eisbedingungen	91
13.2.5	Simulation der realen Voith Wassertrecker (VWT) Rumpfgeometrie	101
14	Abschließende Bemerkungen	104
	Literatur	105

Teil I

Kurzdarstellung des Teilvorhabens

1 Zielsetzung und Motivation

Das Vorhaben befasst sich mit der Entwicklung von effizienten Methoden zur Ertüchtigung von Schiffen, die in arktischen Gewässern verkehren. Gegenwärtig befinden sich ca. 25 % der neu entdeckten Öl- und Gasvorkommen unter (zeitweise) eisbedeckten Gewässern in der Nähe der Arktis. Beispiele hierfür sind die Barentssee oder die Ochotskische See nördlich der Sachalin Inseln. Ferner sind die vormals ganzjährig zugefrorenen arktischen Gewässer aufgrund nachhaltiger klimatischer Veränderungen zusehends weniger lange, weniger weiträumig und mit weniger starkem Eis bedeckt, womit sich die Möglichkeiten zur Erschließung arktischer Schifffahrtsrouten, wie exemplarisch in den Abbildungen 1 und 2 gezeigt, verbessern.

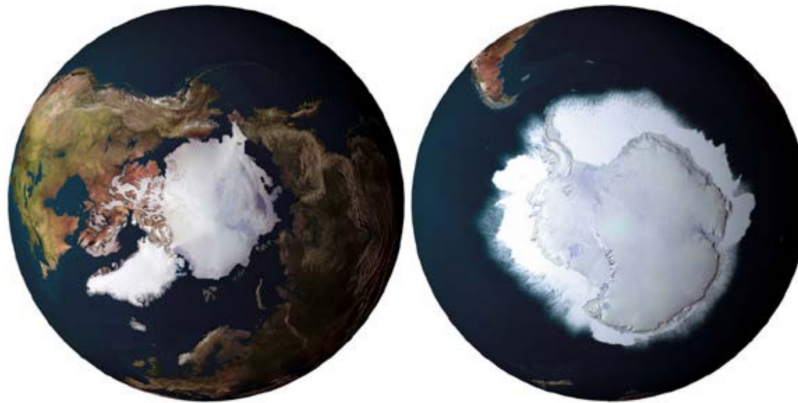


Abbildung 1: Vergleichende Satellitenaufnahmen zur Eisbedeckung der Arktis (links) und Antarktis (rechts) [34].

Die Herausforderungen zur Ertüchtigung von Schiffen und meerestechnischen Bauwerken für arktische Bedingungen sind die sehr großen Umweltrisiken bei technischem Versagen, die sehr großen Eislasten und die schwer zu befriedigenden Positionieranforderungen. In Bezug auf die simulationsbasierte Entwicklung von Designertüchtigungen oder allgemeineren Richtlinien ist die Komplexität des Wechselspiels von Fluid, Eis und Schiff bzw. Propeller der schwierigste Aspekt.

Die Antriebseigenschaften eisgehender Schiffe werden durch ihren Widerstand in Eis, die installierte Leistung und die Propeller-Eis-Interaktion bestimmt. Die Eiszufuhr zum Propeller ist nicht nur für die Propulsionsgüte sondern auch für die Festigkeit des Propellers entscheidend. Sie wird durch die Rumpfform des Schiffes bestimmt, welche die Umströmung des Körpers mit eisbedecktem Wasser steuert und damit die Reduktion der Propulsionsgüte sowie den Anstieg der Propellerbelastung kontrolliert.

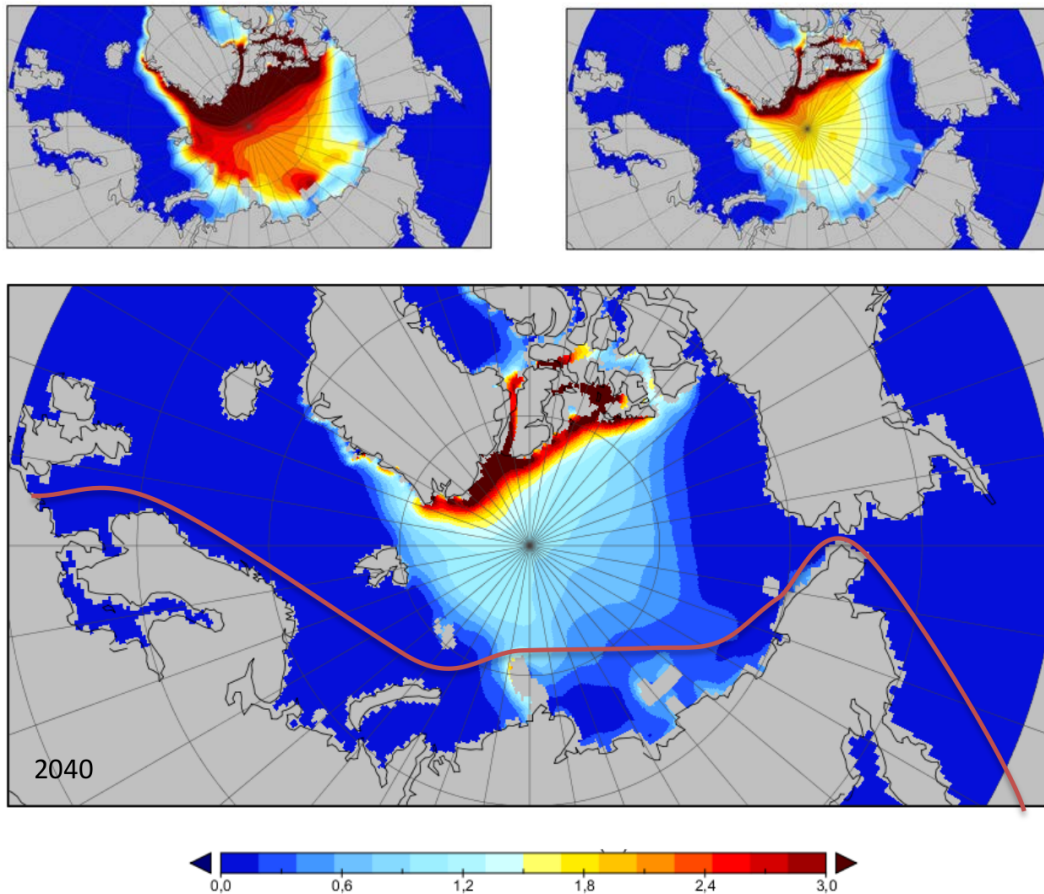


Abbildung 2: Entwicklung der (prognostizierten) minimalen jährlichen Seeeisdicke in Metern. Die gezeichnete Route skizziert den ungefähren Verlauf der Nordostroute von Rotterdam nach Yokohama, die mit ca. 6500sm deutlich kürzer als die ca. 11 200sm lange übliche Suez-Kanal-Route ist (Quelle: AWI).

Die Simulation der Körperumströmung in eisbedecktem Gewässer ist äußerst anspruchsvoll und rechenaufwendig. Sie muss neben der reinen Hydrodynamik auch die Kontaktmechanik zwischen mehreren tausend Eisschollen und dem Rumpf, bzw. zwischen den Eisschollen und dem Propeller berücksichtigen. Da die Wechselwirkung zwischen den vergleichsweise kleinen Eisschollen und dem Arbeitsmedium Wasser aufgelöst werden muss, sind hierzu – anders als bei üblichen CFD-Simulationen – Feldverfahren (Navier-Stokes-Verfahren) mit nahezu isotropen Rechengittern notwendig, deren Zellen nicht wie zumeist praktiziert in Längsrichtung bzw. in Richtung des Fernfeldes stark gestreckt werden können. Daher sind solche Simulationen mit sehr großen Gittern verknüpft, die über mehrere Hunderttausend bzw. Millionen Zeitschritte numerisch integriert werden müssen.

Das Ziel der TUHH ist die Entwicklung eines effizienten Simulationsverfahrens. Hiermit sollen sowohl die Rumpfumströmung in eisbedecktem Gewässer, als auch der Einfluss von Eisdurchtritt durch die Propellerebene auf die Propulsionscharakteristik, simuliert werden können. Grundlage der beantragten Arbeiten ist die Kopplung eines kontaktdynamischen Verfah-

rens zur Erfassung der dynamischen Mehrkörper-Kollisionsprozesse von starren Körpern (engl. *Physics-Engine*) mit einem recheneffizienten Strömungssimulationsverfahren. Hierzu werden ein existierendes Gitter-Boltzmann-Verfahren (efficient lattice boltzmann environment, **elbe** vgl. <https://www.tuhh.de/elbe>) für die Analyse von eisbedeckten Schiffumströmungen sowie eine Physik-Bibliothek gekoppelt, gezielt weiterentwickelt und um weitere physikalische Modelle ergänzt. Das zugrunde liegende **elbe**-Verfahren ist für die Aufgabenstellung wegen seiner hohen Recheneffizienz sehr gut geeignet. Ein wichtiges Merkmal des Verfahrens ist seine Grafikkarten-Fähigkeit. Dadurch lassen sich die vielen tausend Prozessorkerne einer vergleichsweise preiswerten Grafikkarte (GPU) nahezu verlustfrei parallel nutzen. Das Verfahren eignet sich gut bei Strömungssimulationen um komplexe bewegte Geometrien, wie z.B. den relativ zum Schiffsrumpf bzw. den Eisschollen drehenden Propeller. Aufgrund der großen Rechengitter und der begrenzten Speicherkapazität von Grafikkarten sind die Simulationsaufgaben speicherplatzlimitiert. Verbesserungen des Speichermanagements sind daher eine ergänzende Komponente des TUHH-Antrags. Für die Kopplung wird auf die frei verfügbare, quelloffene *Open Dynamics Engine* (ODE) zurückgegriffen. Diese zeichnet durch geringe Nutzungsrestriktionen, einen hohen Verbreitungsgrad (bzw. die daraus zu erwartenden allg. Weiterentwicklungen) sowie die guten Möglichkeiten der Datenstruktur zur effizienten Kollisionsbehandlung auf der Basis von Dreiecksnetzen aus. Sie muss jedoch in Hinblick auf die physikalische Güte der implementierten Modelle und die Effizienz verbessert und angepasst werden. Das Verbundvorhaben war in vier Arbeitspakete gegliedert, wobei die TUHH in den Arbeitspaketen 1, 2 und 3 aktiv war, vgl. Tab. 1.

Tabelle 1: Arbeitspakete im Verbundprojekt PROEIS und deren involvierte Teilnehmer.

Arbeitspaket	Titel
AP 1	Einfluss der Rumpfform eisbrechender und eisgehender Schiffe auf die Propeller-Eis Interaktion (Voith, MMG, MVW, TUHH, HSVA)
AP 2	Einfluss der Propeller-Eis Interaktion auf die Propulsionseffizienz (Voith, MMG, MVW, TUHH, HSVA)
AP 3	Lasten am Propeller verursacht durch Propeller-Eis Interaktion (Voith, MMG, DNVGL, dev, TUHH, HSVA)
AP 4	Großausführungsmessfahrten (MVW, dev, HSVA)

Die Teilarbeitspakete der TUHH stellten hierbei im Rahmen des Verbundprojektes die Entwicklung eines effizienten gekoppelten Simulationsverfahrens (AP 1), die Weiterentwicklung der Physik-Bibliothek (AP 2) sowie die Entwicklung eines Propulsionsmodells (AP 3) dar. Des Weiteren gab es seitens der TUHH die Teilarbeitspakete der Anpassung an den Voith-Schneider-Propeller (VSP) sowie der Anwendung des entwickelten Verfahrens.

2 Ausgangssituation

2.1 Entwurfsmethoden auf der Basis viskoser Hydrodynamik

Der schiffbauliche Designprozess stützt sich vermehrt auf Simulationsverfahren. Neben der Verwendung von klassischen Potenzialmethoden werden heute auch Verfahren zur Analyse turbulent strömender, viskoser Fluide auf der Grundlage der Reynolds-gemittelten Navier-Stokes-Gleichungen (RANS) eingesetzt. Diese Verfahren unterstützen die Berücksichtigung von viskosen und turbulenten Effekten, die z.B. für das Positionierungsverhalten, die Rolldämpfung, Stoppmanöver, die Umströmung des Hinterschiffs, den Widerstand, die Formoptimierung zur Verbesserung der Propulsion, Kavitation oder die Interaktion getauchter Körper in Seegang von Bedeutung sind. Beispiele hierfür finden sich unter anderem in vielen jüngeren Verbundforschungsvorhaben, die vom BMWi (z.B. TUG-Design, Form-Pro, ROLLEX, OptiStop, KonKav, etc.) bzw. der EU (TARGETS, STREAMLINE) gefördert wurden. Hierzu werden gegenwärtig eine Vielzahl von kommerziellen und nicht-kommerziellen Verfahren auf der Grundlage der Finite-Volumen-Approximation eingesetzt (z.B. Fluent, CFX, CCM+, OpenFOAM, FINE-Marine, FreSCo+). Die Mehrzahl der Anwendungen bezieht sich auf quasi-stationäre Anwendungen der Navier-Stokes-Gleichungen auf einen Körper bzw. ein Schiff und ist CPU-basiert. Im Unterschied hierzu soll in dem beantragten Verfahren ein Gitter-Boltzmann-Verfahren eingesetzt werden. Hierbei handelt es sich um einen vergleichsweise neuen, algorithmisch interessanten Weg zur Simulation instationärer Strömungen, dessen wesentliche Merkmale die hohe parallele Effizienz und die sehr guten Möglichkeiten zur Verwendung von energie- und kosteneffizienter GPU-Hardware sind. Die kontinuierliche Steigerung der verfügbaren Rechenleistung dieser Grafikprozessoren und die hohe Effizienz moderner GPU-basierter Algorithmen machen mittlerweile auch im komplexen industriellen Umfeld den Einsatz von Navier-Stokes (NS) Verfahren auf der Grundlage der Lattice-Boltzmann Ansätzen (LB-Verfahren) attraktiv. Im Zusammenhang mit maritimen Aufgabenstellungen sind jedoch nur sehr wenige LB-Applikationen bekannt. Durch die hohe Effizienz der LB-Verfahren sollen in diesem Vorhaben industrielle Effizienzansprüche mit anspruchsvollen Genauigkeitsanforderungen erstmals so kombiniert werden, dass der Aufwand praktischen Anforderungen genügt.

2.2 Entwurfsmethoden für den Verkehr in eisbedeckten Gewässern

Für die industrielle Praxis existiert eine Reihe von Regeln zur Ertüchtigung von Schiffen, die in eisbedeckten Gewässern verkehren, z.B. die Finnish-Swedish Ice Class Rules (FISCR, 1971) oder der IMO Polar Code. Die Formeln besitzen eine empirische Basis und liefern eine direkte Berechnungsgrundlage für die wesentlichen Auslegemerkmale dieser Schiffe, beispielsweise die eisbedingte Antriebsmehrleistung oder die erforderliche (höhere) Materialstärke von Propellerflügelschnitten. Aufgrund ihres empirischen Charakters sind die Regeln nur bedingt bzw. weniger gut zur Extrapolation auf aktuell und zukünftig nachgefragte als auch neue oder größere Schiffstypen geeignet. Sie führen ferner häufig zu sehr konservativen Designvorgaben mit

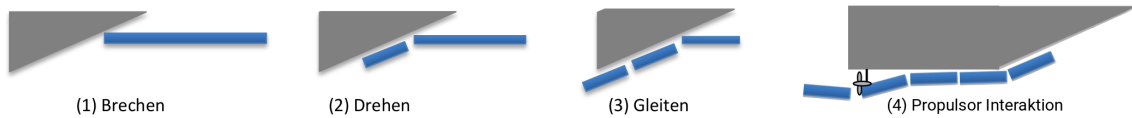


Abbildung 3: Betrachtete Phasen des Eisbrechens sowie der Interaktion der Eisschollen mit dem Schiffsrumpf und dem Propulsor.

großen konstruktiven Folgeproblemen, vgl. Lee [65]. Hieraus ergibt sich ein Bedarf für rationale simulationsbasierte Techniken zur zuverlässigen Analyse von Ertüchtigungsmaßnahmen.

Der Gesamtprozess der Schiff-Eis-Interaktion setzt sich aus vier wesentlichen Phasen zusammen: (1) dem Brechen der Eisschollen, (2) der Drehung der gebrochenen Eisschollen tangential zum Rumpf und (3) dem Verdrängen der Eisschollen unter den Bug bzw. unter das ungebrochene Eis oder in den stromab liegenden eisfreien Kanal, vergleiche Abb. 3. Im Anschluss daran tritt das Eis unter Umständen in Wechselwirkung mit dem Propulsor (4), was hinsichtlich erforderlicher Ertüchtigungsmaßnahmen als durchaus relevant eingestuft werden und insbesondere die im vorliegenden Forschungsvorhaben adressierte Propulsionseffizienz signifikant beeinflussen kann. Im beantragten Vorhaben wird eine globale Simulationsstrategie verfolgt, wobei die hier angestrebte Navier-Stokes-Simulation keine Modellierung der numerisch aufgelösten hydrodynamischen Lasten (z.B. keine virtuellen Fluidmassen) benötigt. Diese gekoppelte Behandlung der Bewegung von Oberflächeneis mit der turbulenten Schiffsumströmung ist neu und geht weit über den Stand der Forschung und Technik hinaus. Ebenso geschieht die Modellierung der Propulsion in eisbedecktem Gewässer aufgrund methodischer Lücken weitestgehend empirisch. Dies soll durch die hier gemeinsam mit der HSVA entwickelten numerischen Verfahren verbessert werden. Die Bewegung der Eisschollen wird dabei unter Zuhilfenahme einer Physik-Bibliothek prognostiziert. Die Bibliothek berücksichtigt in der Bewegungs-Prognose die Schollen-Trägheitskräfte, die hydrodynamischen Lasten, die aus Scholle-Scholle-Interaktionen resultierenden Lasten sowie die Interaktion zwischen den Eisschollen und dem Schiffsrumpf.

Erste Erfahrungen mit dem Einsatz von Physik-Bibliotheken für Eissimulationen waren bereits vor Projektbeginn in der Literatur zu finden (Lubbad und Loset [67]; Metrikin *et al.* [72]; Metrikin *et al.* [73]). In der zitierten Literatur wurde deutlich darauf hingewiesen, dass aktuell verfügbare Ansätze – z.B. die quelloffene Bibliothek *Open Dynamics Engine* [88], die frei verwendbare Nvidia-Bibliothek *PhysX* [81] oder die kommerziellen Tools *Vortex* [22] und *AgX* [8] – zum Teil Probleme bei der Integration von Bewegungsgleichungen besaßen als auch deutliche Schwächen bei der Kollisionsdetektion und Behandlung offenbarten. Die hiermit verknüpften Fehler führten zu erheblichen Körperpositionsabweichungen und signifikanten Verletzungen von Erhaltungsprinzipien und sollen daher hier verbessert werden. Es blieb unklar, welchen Einfluss die nicht aufgelösten Phänomene (z.B. Ventilation) besaßen. Analoge Arbeiten wurden von Daley *et al.* [27, 28] veröffentlicht. Diese nutzten die Möglichkeiten der GPU zur vereinfachten Darstellung der Rumpfbelastung bei Verkehr in Packeis. Die berücksichtigten Eisschollen wurden aus Satellitenaufnahmen rekonstruiert und wirklichkeitsgetreu in einer jedoch nur zweidimensiona-

len Betrachtung angeordnet, in der die Schollen z.B. nicht unter den Rumpf drehen und gleiten oder weiter zerbrechen konnten, vgl. Abb. 3. Das inelastische Kontaktkraftmodell war stark vereinfacht (z.B. reibungsfrei, und parametrisiert), lieferte jedoch sehr gute Eiszusatzwiderstände im Vergleich zu den Arbeiten von Muggeridge und Aboulazm [76], welche ein analytisches Modell formuliert hatten. Der Vergleich mit Arbeiten zu einem mit empirischen Daten entwickelten Modell, welches 3D Effekte (Drehen, Gleiten, Reibung; vgl. Woolgar und Colbourne [109]) offenbarte jedoch Abweichungen in Größenordnung von zum Teil mehreren 100 %.

2.3 Propulsion in Eis und Propeller-Eis-Interaktion

Die Modellierung der Propeller-Eis-Interaktion wird recht gut in den Übersichtsartikeln von Jussila und Soininen [58] sowie Veitch [105] wiedergegeben. Veitch [104] und Soininen [90] sowie Wang [107] setzten die diesbezüglichen Modellierungsarbeiten zur Berücksichtigung von Eis auf die Propulsion und die Propellerbelastung fort.

Veitch [104] legte in seiner Arbeit besonderes Augenmerk auf die Entwicklung eines Kontaktkraftmodells zur Beschreibung des Eisbrechens. Dieses basierte auf der Bestimmung eines Schnitt- bzw. Kollisionswinkels sowie der dazugehörigen Kontaktfläche zwischen der Eisscholle und dem Propeller. Das Modell kombinierte diese Informationen mit analytischen Geometriebeschreibungen zu einer ebenfalls analytischen Hertzschen Kontaktkraft. Die Eisgeometrie beschränkte sich auf Kugeln und die Dynamik wurde an einigen Stellen unterdrückt, z.B. wurde das Eis bei Kontakt nicht deformiert bzw. gespalten und die einzelnen Flügel wurden unabhängig voneinander analysiert. Soininen [90] entwickelte ein Modell zum Bruch des Eises (*Milling*) und formulierte auf der Grundlage von experimentellen Beobachtungen einen Ansatz zur weiteren Behandlung des Propellers beim Durchtritt durch das gebrochene Eis. Der Ansatz behandelte die Druckbelastung bei diesem Vorgang mit Hilfe eines Granularmodells, unterdrückte jedoch hydrodynamische Lasten und die Dynamik zwischen den beteiligten Körpern.

Wang [107] hatte in seiner Dissertation ein weiteres (Rechen-)Modell zur Berücksichtigung von Eis auf die Propulsion und die Propellerbelastung vorgestellt. Das Modell strebte die Berücksichtigung von hydrodynamischen und kontaktdynamischen Kräften. Es gründete auf einem Separationsansatz und überlagert unterschiedliche Phänomene, deren Wechselwirkungen aber unterdrückt wurden. Die von Wang vorgeschlagene Separation unterschied zwischen (1) hydrodynamischen Freifahrteffekten bzw. lasten, (2) Eiseinwirkungen auf die Zuströmung und die damit verknüpften Variation der hydrodynamischen Effekte/Lasten – z.B. Reduktion der effektiven Fortschrittsziffer durch Versperrung der gemittelten Zuströmung (*Blockage*) oder Kavitation durch lokale Beschleunigung (*Proximity*) – sowie (3) Kontaktkräften zwischen dem Eis und dem Propeller (*Milling*). Obwohl konstruktive Details, wie z.B. das Pod-Gehäuse oder die Nabe, vernachlässigt wurden, war das Modell von Wang auf den betrachteten Antriebstyp (Azi-Pod, SPP, Mermaid etc.) zugeschnitten. Dieses grundsätzliche Problem führte bei Zykloldalpropellern dazu, dass nur sehr wenige gut validierte Berechnungsansätze existierten.

Bisherige Untersuchungen – vgl. Wang [107] – an Schraubenpropellern zeigten, dass die hy-

drodynamischen Lasten (Schub, Drehmoment) etwa dem 1,5–2 fachen der Freifahrtsituation entsprachen und die Blattbiegemomente etwa um den Faktor 3 stiegen. Gleichzeitig könnten die lokalen Lasten und die Wechselbelastung stärker steigen. Abhängig vom geometrischen Anstellwinkel traten bei geringen Fortschrittsziffern Abschattungseffekte in Bezug auf Milling-Effekte auf. Für Voith-Schneider- oder Zykloidalpropeller gab es wenig öffentlich verfügbare Erfahrung. Mit dem eingangs erwähnten Hinterschiffmodell sollte es in diesem Vorhaben gelingen, sowohl die hydrodynamischen Versperrungseinflüsse als auch die kontaktdynamischen Effekte (*Milling*) einschließlich des Abschattungsaspektes direkt zu behandeln. Die neu entwickelten Methoden wurden nach der Entwicklung für Schraubenpropeller für Zykloidalpropeller angepasst und abschließend auf zwei repräsentative Testfälle angepasst.

3 Planung und Ablauf

Das Teilprojekt MOUSE der Technischen Universität Hamburg besaß ursprünglich eine Projektlaufzeit von 3 Jahren (01.04.2015 - 31.03.2018). Das Verbundprojekt sollte zudem ursprünglich in 3 Jahren bearbeitet werden und zum 31.03.2018 abgeschlossen sein. Aufgrund einer späteren Personaleinstellung verschob sich der tatsächliche Projektbeginn um 3 Monate auf den 01.07.2015 und wurde – in Absprache mit dem Projektträger sowie den beteiligten Projektpartnern nach einer kostenneutralen Verlängerung um ein Quartal, an der einige Projektpartner teilgenommen haben (HSVA, DNVGL, Voith und TUHH) – final am 30.09.2018 abgeschlossen. Das Teilvorhaben der TUHH wurde bis zum 30.09.2018 kostenneutral verlängert und durch Eigenmittel aufgestockt.

Die Abb. 4 zeigt die ursprüngliche Planung des MOUSE-Projektes in insgesamt 14 Teilarbeitspaketen. Die Abb. 5 zeigt die an die Laufzeitverlängerung angepasste Planung des TUHH-Projektes.

AP	AP Titel	Q216	Q316	Q416	Q117	Q217	Q317	Q417	Q118	Q218
1,1	Schnittstelle/Kopplung PE-Fluid Löser									
1,2	Optimierung für Vielkörperkopplung	2	1							
1,3	Einfaches Propulsionsmodell	2	2							
1,4	Zonale Diskretisierung des Fluidlösers	0,5	1	2	2					
1,5	Speicher- & Laufzeitoptimierung Fluidlöser	0,5								
1	Topic 1	5	4	2	2	0	0	0	0	0
2,1	Verbesserte Kollisionsdetektion Eislöser	3								
2,2	Modellierung nicht-aufgelöster Phänomene		3	3						
2,3	Eiszu-/abfuhr Schnittstelle				2	2	1,5	0,5		
2,4	Eis-/Propellerschnittstelle				1	1	1,5	0,5		
2	Topic 2	3	3	3	3	3	3	1	0	0
3,1	Entwicklung eines Millingverfahrens			1	1	3	1			
3,2	Analyse des aufgelösten Propellers						2	4	4	
3,3	Entwicklung Aktuatorscheibenmodell							1	2	2
4	Anwendung									4
5	Anpassung VSP	1	1,5	1,5	2	2				
Summe		9	8,5	7,5	8	8	6	6	6	6

Abbildung 4: Ursprüngliche Projektplanung des TUHH Teilvorhabens MOUSE.

AP	AP Titel	Q216	Q316	Q416	Q117	Q217	Q317	Q417	Q118	Q218	Q318
1.1	Schnittstelle/Kopplung PE-Fluid Löser										
1.2	Optimierung für Vielkörperkopplung	2	1								
1.3	Einfaches Propulsionsmodell	2	1			1					
1.4	Zonale Diskretisierung des Fluidlösers	0,5	1	2	2						
1.5	Speicher- & Laufzeitoptimierung Fluidlöser	0,5									
1	Topic 1	5	3	2	2	1	0	0	0	0	0
2.1	Verbesserte Kollisionsdetektion Eislöser	3									
2.2	Modellierung nicht-aufgelöster Phänomene				1					1	4
2.3	Eiszu-/abfuhr Schnittstelle				2	2	1,5	0,5			
2.4	Eis-/Propellerschnittstelle					2	2	1	1		
2	Topic 2	3	0	0	3	2	3,5	1,5	1	1	4
3.1	Entwicklung eines Millingverfahrens			1	1	3	1				
3.2	Analyse des aufgelösten Propellers						2	4	4		
3.3	Entwicklung Aktuatorscheibenmodell							2	2	1	
4	Anwendung									2	2
5	Anpassung VSP	1	1,5	1,5		1			1	2	
Summe		9	4,5	4,5	6	7	6,5	7,5	8	6	6

Abbildung 5: Angepasste Projektplanung – kostenneutrale Verlängerung seitens der TUHH.

Zur Gewährleistung einer guten Zusammenarbeit und eines regen Austausches der erzielten Zwischenergebnisse wurden ein Kick-Off-Meeting sowie regelmäßige Projekttreffen organisiert, vgl. Tab. 2.

Tabelle 2: Ergebnisaustausch und Erfolgskontrolle im Verbundvorhaben PROEIS im Rahmen von einem Kick-Off-Meeting, fünf Projekttreffen (PT) sowie einem Workshop im Rahmen der *Ship Efficiency Conference* ausgerichtet von der *Schiffbautechnischen Gesellschaft* (STG).

	Kick-Off	PT-1	PT-2	PT-3	PT-4	Workshop	PT-5
Datum	22. Jul. 2015	26. Jan. 2016	06. Okt. 2016	07. Feb. 2017	27. Jul. 2017	26. Sept. 2017	16. Jan. 2018
Gastgeber	HSVA	HSVA	TUHH	HSVA	HSVA	STG	HSVA

Die wissenschaftlichen Resultate wurden auf sechs nationalen und internationalen Tagungen präsentiert und in begleitenden Tagungsbänden publiziert:

1. Mierke, D., Janßen, C.F., Rung, T., 2015. “GPU-accelerated large-eddy simulation of ship-ice interactions”, VI International Conference on Computational Methods in Marine Engineering, MARINE 2015, Rom.
2. Myland, D., Mierke, D., 2016. “Methodology for Investigation of Ice Breaking Performance”, in: Proc. of the 26th Conference of The International Society of Offshore and Polar Engineers, ISOPE 2016, Rhodos.
3. Mierke, D., Janßen, C.F., Rung, T., 2016. “Efficient calculation of sub-grid distances for higher-order boundary conditions in LBM”, 13. International Conference for Mesoscopic Methods in Engineering and Science, ICMMES 2016, Hamburg.
4. Janßen, C.F., Mierke, D., Rung, T., 2017, “On the development of an efficient numerical ice tank for the simulation of fluid-ship-rigid-ice interactions on graphics processing units”, in: Computers & Fluids.

5. Mierke, D., Gehrke, M., Janßen, C.F., Rung, T., “Validation of a GPU-accelerated fully viscous numerical ice tank”, VII International Conference on Computational Methods in Marine Engineering, MARINE 2017, Nantes.
6. Gehrke, M., Mierke, D., Janßen, C.F., Rung, T., “GPU-Accelerated LBM-VOF Two-Phase Flow Simulations with Grid Refinement”, VII International Conference on Computational Methods in Marine Engineering, MARINE 2017, Nantes.
7. Mierke, D., Janßen, C.F., Rung, T., 2018. “An efficient algorithm for the calculation of sub-grid distances for higher-order LBM boundary conditions in a GPU simulation environment”, in: Computers & Mathematics with Applications.
8. Gehrke, M., Rung, T., Janßen, C.F., 2017. “Scrutinizing lattice Boltzmann methods for direct numerical simulations of turbulent channel flows”, in: Computers & Fluids.
9. Hartmann, M.C.N. 2018. “Entwicklung eines numerischen Simulationsverfahrens zur Berechnung von Eislasten für den Schiffsverkehr in gebrochenem Eis”, Vortrag: STG-Sprechtage „Students meet Industry“ und Veröffentlichung im Jahrbuch der Schiffbautechnischen Gesellschaft.
10. Janßen, C.F. 2018. “GPU-gestützte numerische Simulation von Mehrphasenströmungen im Schiff- und Automobilbau”, Vortrag: 13. Tagung Fahrzeug-Aerodynamik (Haus der Technik), München.
11. Mierke, D., Huisman, M., Gehrke, M., Janßen, C.F., Rung, T., 2017. “Fast prediction of ship-ice interactions with GPUs”, Poster, Graphics Technology Conference (GTC), München.

4 Stand der Forschung vor Projektbeginn

Zum Zeitpunkt des Projektbeginns existierten voneinander unabhängige Versionen des Strömungslösers *elbe* sowie des Starrkörperlösers *Open Dynamics Engine*. Es gab keine Kopplung der beiden Softwarebibliotheken. Bezüglich des Strömungslösers lagen keinerlei funktionale Anpassungen an maritime eisspezifische Anwendungen vor. Zu den relevantesten vor Projektstart fehlenden oder zu optimierenden Modulen zählen insbesondere die Implementierung einer Schnittstelle zwischen *elbe* und *Open Dynamics Engine* sowie einer Eiszufuhrschnittstelle/-mimik, die Modellierung der Interaktion des Strömungsfeldes mit Propulsor, die Propulsorkinematik des Voith-Schneider-Propellers, die Optimierung Hydrodynamik der freien Oberfläche sowie der Durchtritt von Starrkörpern über Gebietsgrenzen unterschiedlich diskretisierter Rechengitter. Üblicherweise werden für numerische Simulationen gekoppelter Verfahren partitionierte Ansätze verfolgt. Diese sind jedoch meist sehr aufwendig und gehen mit Rechenzeiten von Monaten einher. Das Ziel der vorliegenden Forschungsarbeit besteht in der Implementierung eines monolithischen Ansatzes, der eine enge Kopplung der zwei Softwaremodule zum Inhalt hat. Mittels

des monolithischen Ansatzes sowie der Verwendung eines GPU-beschleunigten Strömungslösers wird das Problem der numerischen Simulation von Schiffs-Eis-Interaktionen in einer möglichst effizienten Art und Weise adressiert. Hierbei sind Rechenzeiten in der Größenordnung von Tagen statt Monaten im Falle klassischer Methoden zu realisieren.

Mit dem Beginn der Arbeiten an PROEIS wurde von der TUHH wissenschaftliches Neuland betreten.

5 Zusammenarbeit mit anderen Forschungsstellen

Da die Projektpartner über die zur Durchführung der Arbeitspakete erforderlichen wissenschaftlichen und technischen Kompetenzen verfügten, konnte auf Mithilfe dritter Stellen außerhalb des PROEIS-Konsortiums verzichtet werden.

Teil II

Eingehende Darstellung des Teilvorhabens

1 Einleitung

Das vorliegende Forschungsprojekt befasst sich mit der Entwicklung innovativer hydrodynamischer Entwurfsmethoden für eisgehende Schiffe zur Untersuchung und Optimierung der Propulsionseffizienz. Der dazugehörige derzeitige Stand der Forschung wurde bereits in Kapitel I-2 diskutiert. Zusammenfassend lässt sich festhalten, dass ein dringender Bedarf für rationale simulationsbasierte Techniken zur zuverlässigen Analyse der Schiffsbewegung in eisbedeckten Gewässern festgestellt werden konnte, der im Rahmen dieses Forschungsvorhabens adressiert wurde.

In den initialen Projektphasen wurde zunächst eine grundlegende Kopplung eines effizienten Simulationstools zur Vorhersage der turbulenten Schiffsumströmung an eine Physik-Bibliothek zur Prognose der Bewegung von schollenförmigem Oberflächeneis etabliert. Diese gekoppelte Behandlung der Bewegung von Oberflächeneis mit der turbulenten Schiffumströmung ist neu und geht weit über den Stand der Forschung und Technik hinaus. Phänomenologisch sind hierbei insbesondere die Phasen des Drehens der planar zugeführten Eisschollen, die Phase des Gleitens der gedrehten Schollen entlang des Rumpfes sowie der anschließenden Phase möglicher Interaktionen mit dem Propulsor besondere Aufmerksamkeit gewidmet. Im Anschluss wurde das gekoppelte Verfahren um maßgeschneiderte Schnittstellen zur Analyse der Propulsionseffizienz eisgehender Schiffe erweitert und abschließend auf verschiedene Eis-/Propulsor-Konfigurationen angewendet. Die assoziierten numerischen Simulationen sind aufgrund der hohen Auflösungsanforderungen in Raum und Zeit sowie der aufgrund von $\mathcal{O}(10^3)$ vorhandenen interagierenden Starrkörpern besonders anspruchsvoll und verlangen optimierte numerische Verfahren und angepasste Hochleistungshardware. Die kontinuierliche Steigerung der verfügbaren Rechenleistung von Grafikprozessoren und die hohe Effizienz moderner GPU-basierter Algorithmen machen mittlerweile auch im komplexen industriellen Umfeld den Einsatz von Navier-Stokes (NS) Verfahren auf der Grundlage der Lattice-Boltzmann Ansätzen (LB-Verfahren) attraktiv. Im Zusammenhang mit maritimen Aufgabenstellungen sind jedoch nur sehr wenige LB-Applikationen bekannt. Durch die hohe Effizienz der LB-Verfahren sollen in diesem Vorhaben industrielle Effizienzansprüche mit anspruchsvollen Genauigkeitsanforderungen erstmals so kombiniert werden, dass der Aufwand praktischen Anforderungen genügt. Kapitel 2 fasst zunächst die Grundlagen des verwendeten hoch-effizienten Strömungslösers zusammen.

Ebenso geschieht die Modellierung der Propulsion in eisbedecktem Gewässer aufgrund methodischer Lücken weitestgehend empirisch. Dies wurde durch die im vorgestellten Forschungsvorhaben gemeinsam mit der HSVA entwickelten numerischen Verfahren verbessert.

Der im Rahmen des Verbundvorhabens zu entwickelnde numerische Eiskanal wurde um eine Eiszufuhrschnittstelle, die Modellierung der Interaktion des Strömungsfeldes mit Propulsor, die

Propulsorkinematik des Voith-Schneider-Propellers, die Optimierung Hydrodynamik der freien Oberfläche sowie der Durchtritt von Starrkörpern über Gebietsgrenzen unterschiedlich diskretisierter Rechengitter erweitert. Grundlagen der jeweiligen Modelle werden in den folgenden Kapiteln ausführlich diskutiert. Durch die Verwendung eines GPU-beschleunigten Strömungslösers sowie einer besonders effizienten Implementierung der erforderlichen Verfahrenserweiterungen ermöglicht der im PROEIS-Verbundforschungsvorhaben entwickelte numerische Eiskanal die numerische Simulation von Schiffs-Eis-Interaktionen mit Rechenzeiten in der Größenordnung von Tagen statt Monaten.

2 Numerische Simulation mit dem Gitter-Boltzmann-Verfahren

Für die numerische Simulation wird im Rahmen dieser Arbeit eine mittels Grafikkarten, engl. *Graphics Processing Unit* (GPU), beschleunigte Implementierung des Gitter-Boltzmann-Verfahrens, engl. *Lattice Boltzmann Method* (LBM), verwendet. Anstatt der Lösung der auf makroskopischen Größen (Dichte und Impuls) basierenden Navier-Stokes-Gleichungen, wird im Kontext der LBM die diskretisierte mesoskopische Boltzmann-Gleichung verwendet, die für einen Satz von Verteilungsfunktionen auf einem regulären, kartesischen Gitter gelöst wird. Es kann gezeigt werden, dass für ausreichend kleine Raum- und Zeitschritte sowie kleine Mach- und Knudsen-Zahlen das Gitter-Boltzmann-Verfahren bis auf Fehlerterme zweiter Ordnung $-\mathcal{O}(\Delta t^2)$, $\mathcal{O}(\text{Ma}^2)$ sowie $\mathcal{O}(\text{Kn}^2)$ – zur Lösung der im makroskopischen Kontext definierten inkompressiblen Navier-Stokes-Gleichungen konvergiert.

Die Boltzmann-Gleichung stellt die substantielle Ableitung der Verteilungsfunktion, engl. *Particle Distribution Function* (PDF), eines auf einer mesoskopischen Skala definierten fiktiven Partikels dar. Die Verteilungsfunktion $f(\mathbf{x}, t, \boldsymbol{\xi})$ gibt die Wahrscheinlichkeit des Auffindens eines Partikels an der Position \mathbf{x} zur Zeit t mit der (Partikel-) Geschwindigkeit $\boldsymbol{\xi}$ an [83, 110]. Die von Ludwig Boltzmann aufgestellte Boltzmann-Gleichung lautet

$$\frac{\partial f(\mathbf{x}, t, \boldsymbol{\xi})}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f(\mathbf{x}, t, \boldsymbol{\xi})}{\partial \mathbf{x}} = \Omega. \quad (1)$$

Die linke Seite der Boltzmann-Gleichung ist vom Advektionstyp und beschreibt die instationäre Änderung, den konvektiven Partikeltransport sowie den Einfluss äußerer Kräfte. Auf der rechten Seite der Gleichung tritt der Kollisionsoperator Ω auf, der die Partikelinteraktionen beschreibt.

2.1 Grundgleichungen des Gitter-Boltzmann-Verfahrens

Für Strömungen, die eine niedrige Knudsen-Zahl Kn aufweisen und demnach die Kontinuums-hypothese befriedigen, kann die kontinuierliche Verteilungsfunktion der Boltzmann Gleichung (1) bezüglich eines diskreten Geschwindigkeitsraumes mit den Geschwindigkeiten \mathbf{e}_{ijk} approximiert werden, um ein Modell reduzierten Rechenaufwandes zu erhalten. Mit dieser diskretisierten Formulierung geht einher, dass sich ein Partikel von einem gegebenen Gitterpunkt aus

nur in einer begrenzten Anzahl von Richtungen und für bestimmte Abstände bewegen kann. Unter Verwendung dieser Approximationen geht die Boltzmann-Gleichung (1) in die diskrete Boltzmann-Gleichung

$$\frac{\partial f_{ijk}(\mathbf{x}, t)}{\partial t} + \mathbf{e}_{ijk} \cdot \frac{\partial f_{ijk}(\mathbf{x}, t)}{\partial \mathbf{x}} = \Omega_{ijk} \quad (2)$$

über, worin $\mathbf{e}_{ijk} = c \times (i, j, k)$ und $i, j, k \in (-1, 0, 1)$ sind. Die Gittergeschwindigkeit ist als Quotient der Gitter- (Δx) und Zeitschrittweite (Δt) definiert, $c = \Delta x / \Delta t$. Der Wert für c wird im Allgemeinen auf eins gesetzt, wodurch festgelegt ist, dass sich ein Partikel innerhalb eines diskreten Zeitschrittes um exakt einen Gitterabstand zu den umgebenden Nachbarknoten fortbewegt. Die eingeführte Definition von \mathbf{e}_{ijk} beschreibt ein dreidimensionales Gitter mit 19 diskreten Geschwindigkeitsvektoren, das als *D3Q19*-Modell bezeichnet wird. Die Notation beinhaltet sowohl die Dimension (dreidimensional) als auch die Anzahl der diskreten Freiheitsgrade (19), der sogenannten Kollokationspunkte, eines Partikels, vgl. Abb. 6.

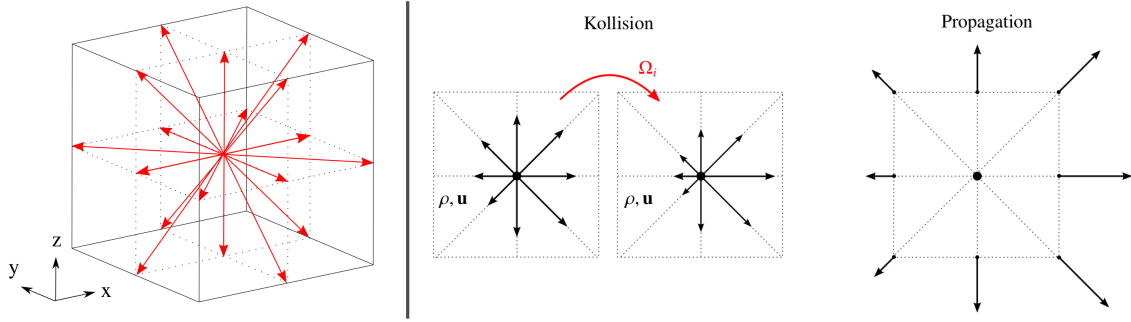


Abbildung 6: Diskreter *D3Q19* Geschwindigkeitsraum, welcher der verwendeten LBM Implementierung zugrunde liegt (links). Schematische Darstellung des Kollisions- und Propagationschrittes im Zweidimensionalen (rechts).

Die numerische Lösung der diskreten Boltzmann Gleichung (2) mittels einer räumlichen und zeitlichen Diskretisierung über finite Differenzen ergibt die Gitter-Boltzmann-Gleichung

$$f_{ijk}(\mathbf{x} + \mathbf{e}_{ijk} \Delta t, t + \Delta t) = f_{ijk}(\mathbf{x}, t) + \Delta t \Omega_{ijk}. \quad (3)$$

Die makroskopischen Größen des Drucks (resp. der Dichte) und des Geschwindigkeitsvektors liegen lokal an jedem Gitterknoten vor und sind als Momente nullter sowie erster Ordnung der Partikelverteilungsfunktionen definiert

$$p = c_s^2 \sum_{i=-1}^1 \sum_{j=-1}^1 \sum_{k=-1}^1 f_{ijk} \quad \rho = \sum_{i=-1}^1 \sum_{j=-1}^1 \sum_{k=-1}^1 f_{ijk} \quad \text{und} \quad (4)$$

$$\mathbf{u} = \frac{1}{\rho} \sum_{i=-1}^1 \sum_{j=-1}^1 \sum_{k=-1}^1 f_{ijk} \mathbf{e}_{ijk}, \quad (5)$$

wobei $c_s = c/\sqrt{3}$ die Schallgeschwindigkeit im Gitter ist.

Die Gleichung (3) wird in der Numerik in zwei getrennten Schritten ausgeführt. Der erste Schritt

ist der sogenannte Kollisionsschritt,

$$\bar{f}_{ijk}(\mathbf{x}, t) = f_{ijk}(\mathbf{x}, t) + \Delta t \Omega_{ijk}, \quad (6)$$

der die lokale Annäherung der an einem Gitterknoten ankommenden Verteilungsfunktionen hin zu einer vorgegebenen Gleichgewichtsverteilung einschließt und somit die Partikelinteraktion beim Aufeinandertreffen modelliert. Hierin beschreibt \bar{f}_{ijk} den Zustand der Partikelverteilungsfunktion nach der Kollision. Die LBM Charakteristik der strikten Lokalität des Kollisionsschrittes stellt einen der größten Vorteile dieser Methode dar und ermöglicht deren parallele Implementierung. Dies geht wiederum mit der perfekten Eignung eines LBM Löser für die effiziente Umsetzung unter Verwendung massiv paralleler GPGPU (engl. *General Purpose Computation on Graphics Processing Unit*) Architekturen auf Hochleistungsrechnern einher. Der zweite Teil der numerischen Lösung ist der Propagationsschritt

$$f_{ijk}(\mathbf{x} + \mathbf{e}_{ijk} \Delta t, t + \Delta t) = \bar{f}_{ijk}(\mathbf{x}, t). \quad (7)$$

Dieser umfasst die Übertragung der Verteilungsfunktionen zu den nächstgelegenen Nachbarknoten.

In dem rechten Teil der Abbildung 6 sind der lokale LBM Kollisionsschritt, der eine knotenbasierte Umverteilung der PDF unter Beibehaltung der makroskopischen Dichte und Impuls als konservative Größen während der Kollision entspricht sowie die Propagation der PDF entlang der diskreten Gitterrichtungen aus Gründen der Übersichtlichkeit schematisch für den Zweidimensionalen $D2Q9$ Geschwindigkeitsraum dargestellt.

2.2 Implementierung

Die Implementierung des Gitter-Boltzmann-Verfahrens ist in dem am Institut für Fluidodynamik und Schiffstheorie (M-8) der Technischen Universität Hamburg eingesetzten Strömungslöser **e1be** (engl. *efficient lattice boltzmann environment*, [52]) umgesetzt. **e1be** wird seit dem Entwicklungsstart im Jahr 2012 fortan laufend weiterentwickelt und um diverse Funktionalitäten erweitert. Der Strömungslöser ermöglicht die Simulation ein-, zwei- sowie dreidimensionaler Problemstellungen. **e1be** ist in unterschiedliche Module aufgeteilt und bietet unter Anderem die Möglichkeit der Simulation von Flachwasser, Ein- und Mehrphasenströmungen sowie Strömungen mit freier Oberfläche. Die Implementierung ist in der Programmiersprache C++ unter der Verwendung der CUDA [26, 32] (engl. *Compute Unified Device Architecture*) Programmierumgebung von Nvidia ausgeführt [80]. Die strenge Lokalität des Gitter-Boltzmann-Verfahrens ermöglicht hierbei die massiv parallelisierte Ausführung auf Grafikprozessoren, die beispielsweise in [51] im Kontext des zweidimensionalen Gitter-Boltzmann-Verfahrens beschrieben wird.

2.3 MRT-Kollisionsoperator

Das LBM-Verfahren besteht aus einem Kollisionsschritt (6) und einem Propagationsschritt (7). In der Literatur gibt es unterschiedliche Techniken variierender Komplexität, numerischer Genauigkeit und numerischer Stabilität zur Formulierung des Kollisionsschrittes. Neben der einfachen Relaxation der Nichtgleichgewichtsanteile des Verteilungsfunktionsvektors $\underline{f}^{\text{neq}} = \underline{f} - \underline{f}^{\text{eq}}$ hin zu einem definierten Equilibriumszustand $\underline{f}^{\text{eq}}$ (lokale Maxwell-Gleichgewichtsverteilung) mit derselben (skalaren) Rate ω (engl. *Single-Relaxation-Time-Modell*, [15]),

$$\underline{\Omega} = -\omega (\underline{f} - \underline{f}^{\text{eq}}) \quad , \quad (8)$$

oder mit zwei Relaxationsraten (engl. *Two-Relaxation-Time-Modell*, [40, 39]), gibt es auch höherwertigere Strategien, in denen anstelle der Verteilungsfunktionen \underline{f} hieraus abgeleitete Größen, die sich im Allgemeinen aus $x_i = X_{ik} f_k$ ergeben, mit individuellen Raten relaxiert werden

$$\underline{\Omega}_{(x)} = -\underline{S}_{\omega} \cdot (\underline{x} - \underline{x}^{\text{eq}}) . \quad (9)$$

Hierin ist \underline{S}_{ω} die Diagonalmatrix der individuellen Relaxationsraten. Das Ergebnis wird dann über eine inverse Transformationsvorschrift ($\underline{f} = \underline{X}^{-1} \cdot \underline{x}$) wieder in den Raum der Verteilungsfunktionen überführt.

Für die numerischen Simulationen im Rahmen des Forschungsprojektes wurde für den Kollisionsoperator auf ein *Multiple-Relaxation-Time-Modell* zurückgegriffen, das in seiner ursprünglichen Form von [30] eingeführt wurde. Die numerische Umsetzung des *MRT* Gitter-Boltzmann Kollisionsoperators ist in der Abb. 7 als Flussdiagramm skizziert.

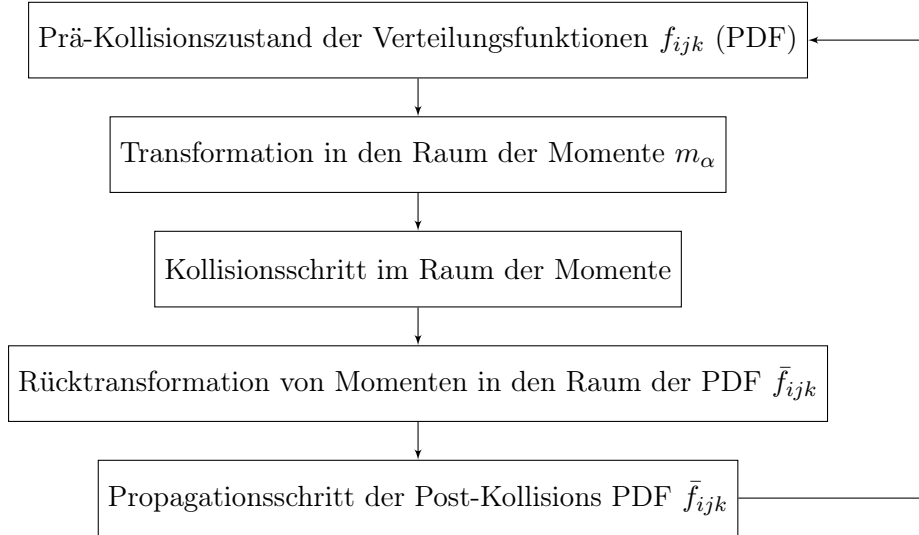


Abbildung 7: Schematischer Ablauf der numerischen Umsetzung des *MRT* Gitter-Boltzmann Kollisionsoperators.

Die Wahl gewichteter statt ungewichteter orthogonaler Basisvektoren zur Erzeugung der Trans-

formationsvorschrift in den Momentenraum [29, 12] führt zu der hier verwendeten *MRT* Formulierung von [100]. Im Vergleich zu dem einfachen *SRT*-Modell weist das *MRT*-Modell eine signifikante Verbesserung der numerischen Stabilität auf, die sich speziell für Strömungen hoher Reynolds-Zahl und damit verbundener geringer Viskosität als vorteilhaft erweist.

Das *MRT*-Modell basiert auf einer Transformation der Verteilungsfunktionen in einen Momentenraum, in dem anschließend die Relaxation hin zu einem definierten Gleichgewichtszustand ausgeführt wird. Der Momentenraum wird mittels Linearkombinationen der PDF aufgespannt. Der Vorteil des *MRT* besteht darin, dass den einzelnen Momenten physikalische Äquivalente zugewiesen werden können, wodurch eine Trennung der Momente nach hydrodynamischen und nicht-hydrodynamischen Größen ermöglicht wird. Diese wiederum können mit voneinander unabhängigen Relaxationsraten relaxiert werden, wodurch sich ein positiver Einfluss auf die numerische Stabilität des Verfahrens ergibt. Die allgemeine Form des *D3Q19 MRT* Kollisionoperators ist durch

$$\Omega_\alpha = - \sum_{\beta=1}^{19} M_{\alpha\beta}^{-1} \left(\sum_{\gamma=1}^{19} S_{\beta\gamma} \left(\left(\sum_{\delta=1}^{19} M_{\gamma\delta} f_\delta \right) - m_\gamma^{\text{eq}} \right) \right) \quad (10)$$

gegeben. Die Matrix $S_{\alpha\beta}$ ist eine Diagonalmatrix, welche die einzelnen Relaxationsraten enthält und $M_{\alpha\beta}$ stellt die lineare Transformationsmatrix dar. Die Transformationsmatrix nach [100] aus dem Raum der Verteilungsfunktionen in den Raum der Momente ist als

$$M_{\alpha\beta} = \begin{bmatrix} 1 \cdot (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ c^2 \cdot (-1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ c^4 \cdot (1 & -2 & -2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ c \cdot (0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0) \\ c^3 \cdot (0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0) \\ c \cdot (0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1) \\ c^3 \cdot (0 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1) \\ c \cdot (0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1) \\ c^3 \cdot (0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1) \\ c^2 \cdot (0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2) \\ c^4 \cdot (0 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2) \\ c^2 \cdot (0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0) \\ c^4 \cdot (0 & 0 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0) \\ c^2 \cdot (0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0) \\ c^2 \cdot (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1) \\ c^2 \cdot (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0) \\ c^3 \cdot (0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0) \\ c^3 \cdot (0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1) \\ c^3 \cdot (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & -1) \end{bmatrix} \quad (11)$$

definiert. Die sich ergebenden Momente werden als

$$m_\alpha = (\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z)^T \quad (12)$$

benannt. Die auf den makroskopischen Strömungsgrößen, der Dichte und des Impulses, basierenden Gleichgewichtsmomente sind als

$$\begin{aligned} m_1^{\text{eq}} &= \rho, & m_2^{\text{eq}} &= e^{\text{eq}} = \rho_0 (u_x^2 + u_y^2 + u_z^2), \\ m_4^{\text{eq}} &= \rho_0 u_x, & m_6^{\text{eq}} &= \rho_0 u_y, & m_8^{\text{eq}} &= \rho_0 u_z, \\ m_{10}^{\text{eq}} &= 3p_{xx}^{\text{eq}} = \rho_0 (2u_x^2 - u_y^2 - u_z^2), & m_{12}^{\text{eq}} &= p_{zz}^{\text{eq}} = \rho_0 (u_y^2 - u_z^2), \\ m_{14}^{\text{eq}} &= p_{xy}^{\text{eq}} = \rho_0 u_x u_y, & m_{15}^{\text{eq}} &= p_{yz}^{\text{eq}} = \rho_0 u_y u_z, & m_{16}^{\text{eq}} &= p_{xz}^{\text{eq}} = \rho_0 u_x u_z, \\ m_3^{\text{eq}} &= m_5^{\text{eq}} = m_7^{\text{eq}} = m_9^{\text{eq}} = m_{11}^{\text{eq}} = m_{13}^{\text{eq}} = m_{17}^{\text{eq}} = m_{18}^{\text{eq}} = m_{19}^{\text{eq}} = 0 \end{aligned} \quad (13)$$

definiert und die Diagonalmatrix, welche die Relaxationsraten enthält, entspricht

$$S_{\alpha\beta} = \text{diag}(0, s_a, s_b, 0, s_c, 0, s_c, 0, s_c, s_\omega, s_d, s_\omega, s_d, s_\omega, s_\omega, s_e, s_e, s_e). \quad (14)$$

Hierin ist $s_\omega = \Delta t/\tau = 1/\tau = \omega$ mit der Scherviskosität ν_0 des zugrundeliegenden Fluides via $\tau = \nu_0/c_s^2 + \Delta t/2$ verknüpft und die weiteren fünf voneinander unabhängigen Relaxationsparameter $(s_a, s_b, s_c, s_d, s_e)$, die zur Relaxation nicht-hydrodynamischer Momente verwendet werden, können frei aus dem Intervall $\in]0, 2[$ gewählt werden.

2.4 Grobstruktursimulation – Smagorinsky Turbulenzmodell

Die im Rahmen dieses Forschungsprojektes simulierte Strömungen entsprechen allesamt instationären, turbulenten Strömungen hoher Reynolds-Zahl $\mathcal{O}(10^7)$. Zur Auflösung aller in einer turbulenten Strömung beteiligten Strukturen sind Knotenanzahlen der Größenordnung $\mathcal{O}(Re^{9/4})$ zu verwenden. Die Erfüllung dieser Bedingung macht die direkte numerische Simulation aus Gründen der verfügbaren Ressourcen, wie Rechenleistung und Speicher, entweder unmöglich oder geht selbst für mäßige Reynolds-Zahlen $\mathcal{O}(10^4 - 10^5)$ mit Rechenzeiten einher, die aus ingenieurmäßiger Sicht ungerechtfertigt hoch und damit für praxisnahe Simulationen inakzeptabel sind. Zur effizienten wie auch genauen Simulation turbulenter Strömungen werden in dem Kontext dieser Arbeit Grobstruktursimulationen (engl. *Large Eddy Simulation* (LES)) ausgeführt. Hierbei werden die großskaligen energietragenden Wirbelstrukturen sowie die kleinskaligen dissipativen Wirbel des turbulenten Energiespektrums durch einen Filter voneinander getrennt. Die Strukturen, die kleinere Ausmaße als die gewählte Filterweite besitzen, müssen demnach modelliert werden.

Zur Modellierung wird hierbei herkömmlicherweise eine gradientenbasierte isotrope Wirbelviskosität ν_t herangezogen, welche der Scherviskosität des Fluides addiert wird: $\nu_{ges} = \nu_0 + \nu_t$. Die gebräuchlichste Formulierung des Schließungsproblems stellt das Smagorinsky-Lilly Funk-

tional [87], $\tau_{im} \sim 2\nu_t S_{im}$ mit $\nu_t = (C_s \Delta x_i)^2 \sqrt{2S_{jk}S_{jk}}$, dar. Dieses gibt die Wirbelviskosität in Abhängigkeit einer Filterweite Δx , des Betrages des Scherratentensors

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right) \quad (15)$$

und der Smagorinsky-Konstanten C_s (üblicherweise $C_s = 0.18$) an. Diese ist für das gesamte Strömungsgebiet gleich und spiegelt somit den isotropen Charakter des Modells wieder. Die phänomenologische Interpretation der Wirbelviskosität entspricht der Modellierung der Dissipation im Bereich der kleinsten Wirbelstrukturen, die aufgrund der (für hohe Re teilweise sehr groben) Unterauflösung des Strömungsproblems nicht direkt abgebildet werden können.

Für das Gitter-Boltzmann-Verfahren wird die Gleichung zur Berechnung der Wirbelviskosität in [68] zu

$$\begin{aligned} \nu_t &= (C_s \Delta x_i)^2 \sqrt{2S_{jk}S_{jk}} \\ &= \frac{3}{2} \frac{1}{\tau_{ges}} (C_s \Delta x) |Q_{ij}| \\ &= \frac{3}{2} \omega_{ges} (C_s \Delta x) \sqrt{\sum_{i,j} Q_{ij} Q_{ij}} \end{aligned} \quad (16)$$

umgeschrieben. In die modifizierte Gleichung geht die durch $\tau_{ges} = \tau_0 + \tau_t = 1/\omega_{ges}$ definierte gesamte Relaxationszeit und der Betrag des Tensors Q_{ij} ein, dessen Einträge sich direkt aus den lokalen Nicht-Gleichgewichtsanteilen der Verteilungsfunktionen,

$$\begin{aligned} Q_{ij} &= \sum_{\alpha} e_{\alpha i} e_{\alpha j} (f_{\alpha} - f_{\alpha}^{eq}) \\ &= \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{neq}, \end{aligned} \quad (17)$$

zusammensetzen, die mit Produkten der diskreten Geschwindigkeitsvektoren e_{α} multipliziert werden. Mittels der Zusammenhänge $\tau_t = 3\nu_t$ sowie $\tau_{ges} = \tau_0 + \tau_t$ ergibt sich aus der Lösung der Gleichung 16 die turbulente Relaxationszeit zu

$$\tau_t = \frac{1}{2} \left(\sqrt{\tau_0^2 + 18C_s^2 \sqrt{\sum_{i,j} Q_{ij} Q_{ij}}} - \tau_0 \right). \quad (18)$$

Die modifizierte zusammengesetzte Relaxationsfrequenz ist abschließend durch

$$\omega_{ges} = \frac{1}{\tau_0 + \tau_t} \quad (19)$$

gegeben.

2.5 Freie-Oberflächen-Modell

Freie Oberflächen unmischbarer Zweiphasenströmungen werden von der dichteren Phase dominiert, wenn beide Phasen sich stark in Dichte und Viskosität unterscheiden. Unter Vernachlässi-

gung des Kapillareffekts und der Oberflächenspannung ist es ausreichend, lediglich die dichtere Phase zu simulieren. Der Einfluss der leichteren Phase lässt sich anschließend durch kinematische und dynamische Randbedingungen an der freien Oberfläche berücksichtigen. Die freie Oberfläche darf sich zwar frei bewegen, muss jedoch zu jeder Zeit so eindeutig und dünn wie möglich gehalten werden.

In der LBM findet zur Simulation freier Oberflächen im Allgemeinen die sogenannte *Volume of Fluid*-Methode (VoF) in Kombination mit einem flussbasierten mesoskopischen Advektionsmodell Anwendung, wie Körner *et al.* [62] sowie Janßen und Krafczyk [55] zeigen. Diese Methode ist aufgrund ihrer direkten und simplen Oberflächenrekonstruktion sowie des zeitlich expliziten Verfahrens, wie die LBM, prädestiniert für die Implementierung auf parallelen Rechnerarchitekturen. Im Gegensatz zu gängigen VoF-Methoden, welche die konvektive Transportgleichung makroskopisch lösen, lässt sich der Flussterm direkt aus den Verteilungsfunktionen f_i bestimmen.

Die VoF-Methode erfasst die freie Oberfläche, auch Interface (Phasentrennfläche) genannt, im Euler-Raum mittels der Füllstandvariable ϵ . Ein Füllstand von $\epsilon = 0$ kennzeichnet eine leere Zelle in der inaktiven bzw. vernachlässigten Gasphase. Hingegen beschreibt $\epsilon = 1$ eine komplett gefüllte Zelle in der Fluidphase. Fluid- und Gasphase sind strikt durch einen Übergangsbereich, dem Interface, mit dem Füllstand $\epsilon \in [0, 1]$ voneinander getrennt, vgl. Abb. 8.

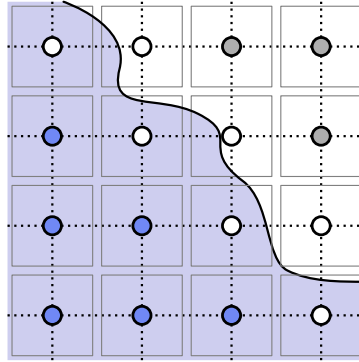


Abbildung 8: Fluid- (blau), Interface- (weiß) und Gaszellen (grau) der VOF-Methode.

Der neue Füllstand einer Zelle ergibt sich durch die Bilanzierung der Massenströme Δm_i zu den benachbarten Zellen:

$$\epsilon^{t+1} = \frac{m^{t+1}}{\rho^{t+1}} = \frac{\rho^t \epsilon^t + \sum_i \Delta m_i}{\rho^{t+1}} \quad \text{mit} \quad \Delta m_i = \Delta t A_i (f_I(t, \mathbf{x}) - f_i(t, \mathbf{x})) . \quad (20)$$

A_i beschreibt die benetzte Fläche zwischen zwei Zellen und berechnet sich vereinfacht aus dem

arithmetischen Mittel der Füllstände beider Zellen:

$$A_i = \begin{cases} 1,0 & : \text{Nachbar ist Fluid,} \\ \frac{\epsilon(t, \mathbf{x}) + \epsilon(t, \mathbf{x} + \Delta t \mathbf{e}_i)}{2} & : \text{Nachbar ist Interface,} \\ 0,0 & : \text{Nachbar ist Gas oder Hindernis.} \end{cases} \quad (21)$$

An der freien Oberfläche gilt eine Druckrandbedingung, die für eine Bilanzierung des Fluid- und Umgebungsdrucks p_B sorgt.

Neue Fluidknoten werden mit Hilfe der diskreten Maxwell'schen Gleichgewichtsverteilung initialisiert:

$$f_i = f_i^{\text{eq}}(\bar{\rho}, \bar{\mathbf{u}}), \quad (22)$$

wobei $\bar{\rho}$ und $\bar{\mathbf{u}}$ aus den benachbarten Fluidknoten interpoliert werden [55]:

$$\bar{\rho}(\mathbf{x}) = \sum_i w_i \rho(\mathbf{x} + \Delta t \mathbf{e}_i) \quad \text{sowie} \quad \bar{\mathbf{u}}(\mathbf{x}) = \sum_i w_i \mathbf{u}(\mathbf{x} + \Delta t \mathbf{e}_i). \quad (23)$$

Die Methode ist im Allgemeinen Massen-erhaltend und stabil, allerdings nur von erster Ordnung im Raum genau, da der Normalenvektor der freien Oberfläche nicht berücksichtigt wird [53]. Diese und weitere Methoden finden sich in Gunstensen *et al.* [43], Ginzburg *et al.* [38] sowie Thürey und Rüde [99]. Eine Kombination aus VoF und Level-Set [82, 96] präsentiert Überrück [103], welche die Vorteile beider Methoden vereint, indem sie sowohl die genaue Position als auch Ausrichtung der freien Oberfläche berücksichtigt.

2.5.1 VoF-Erweiterungen an elbe

Um Tripelpunkte im Bereich der freien Wasseroberfläche zwischen Fluid und Strömungshindernis hinreichend genau erfassen und behandeln zu können, wurde `elbe` an diversen Stellen umfangreich überarbeitet und ausgebaut. Für eine genauere Methode ist dennoch ein Freie-Oberflächen-Modell höherer Ordnung erforderlich, das auf einem ausgereiften Verfahren für die Berücksichtigung beliebig beweglicher Strömungshindernisse basiert.

Insbesondere der Fall, wenn sich zwei eng aneinander liegende Strömungshindernisse in tief- oder teilgetauchter Phase befinden, eine erhöhte Kontaktfläche aufweisen und sich normal zur Kontaktfläche separieren. Bei diesem Problem gilt es, von einem auf den anderen Zeitschritt eine ganze Gitterknotenfläche zu reinitialisieren, die unter Umständen mehrere Gitterknoten breit als auch lang ist und vermehrt Gitterknoten umfasst, die keine validen Fluidnachbarknoten besitzen. Um diesen *Gitterknotenteppich* zu Reinitialisieren, ist es im Allgemeinen erforderlich, komplexe iterative Techniken zu verwenden, die noch im selben Zeitschritt für eine möglichst physikalisch konsistente Reinitialisierung sorgen. Solche Techniken lassen sich für gewöhnlich nur schwer mit einem expliziten Strömungslöser vereinbaren. Daher wurde ein Ansatz gewählt, der den zu reinitialisierenden Bereich, von der validen Berandung nach Innen gehend, Zeitschritt

für Zeitschritt weiter schließt. Das Verfahren wurde erst vor kurzem implementiert und steht noch weiteren Validierungen aus. Dennoch erfüllt es für die bislang ausgeführten Simulationen der Anwendungsfälle (vgl. Kapitel 13) sehr zuverlässig seinen Zweck.

2.6 Gitterverfeinerung – Zonale Methoden des Fluidlösers

Im Rahmen der Arbeiten dieses Forschungsprojektes wurden Algorithmen für die lokale Verfeinerung des Rechengebietes einerseits aus der Literatur adaptiert sowie andererseits umfangreich weiter- und neu entwickelt und in den Strömungslöser `elbe` implementiert. Die in diesem Abschnitt behandelten Gitterverfeinerungstechniken beziehen sich auf konstant zueinander ausgerichtete Gitterkonfigurationen, deren Positionen initial vor dem Programmstart festgelegt werden und zur Laufzeit unveränderlich bleiben. Im Gegensatz hierzu gibt es den Typus der körperfesten Gitter, die sich bei der Bewegung des Körpers dynamisch durch das Rechengebiet mitbewegen. Auf diese Art der zonalen Methode wird detailliert in Abschnitt 12 eingegangen.

Das Gitter-Boltzmann-Verfahren arbeitet auf äquidistanten, kartesischen Gittern. Hierdurch kann Gleichung 7 besonders effizient implementiert und umgesetzt werden. Sind dennoch lokal höhere Gitterauflösungen (respektive feinere Zeit- und Raumschrittweiten) – aufgrund großer zu erwartender Gradienten des Strömungsfeldes, der Notwendigkeit des Erfassens kleiner Strukturen oder zur genaueren Auflösung von Geometrien – in einem bestimmten Bereich der numerischen Simulation erforderlich, können verschiedene LBM-Gitter kombiniert und im Gitterübergangsbereich miteinander verknüpft werden. In Abb. 9 sind zwei charakteristische Szenarien gezeigt in denen Gitterverfeinerungen zur Steigerung der Genauigkeit zum Einsatz kommen:

- Hohe zu erwartende Gradienten der Strömungsgrößen. Die Umströmung des wandgebundenen Quaders benötigt zur Auflösung kleinskaliger Strukturen ein Gebiet feinerer Diskretisierung. Die in das Fernfeld integrierte und überlappende Verfeinerungsregion wird durch den weißen Umriss angedeutet.
- Bessere bzw. genauere Auflösung der durch das Rechengebiet bewegten Geometrien (graue Quader deuten die voxelbasierte Geometrierepräsentation auf dem kartesischen LBM Gitter an). Die feine Diskretisierung (links) sieht eine den gesamten Propulsor (hier Voith-Schneider-Propeller) umgebende, achsenparallele Gitterverfeinerungsbox vor, die im Vergleich zu der groben Diskretisierung – unter Verwendung der uniformen Gitterweite des Fernfeldes für die gesamte numerische Domain – eine deutliche Verbesserung der Geometrieabbildung sowie des resultierenden Strömungsfeldes auf.

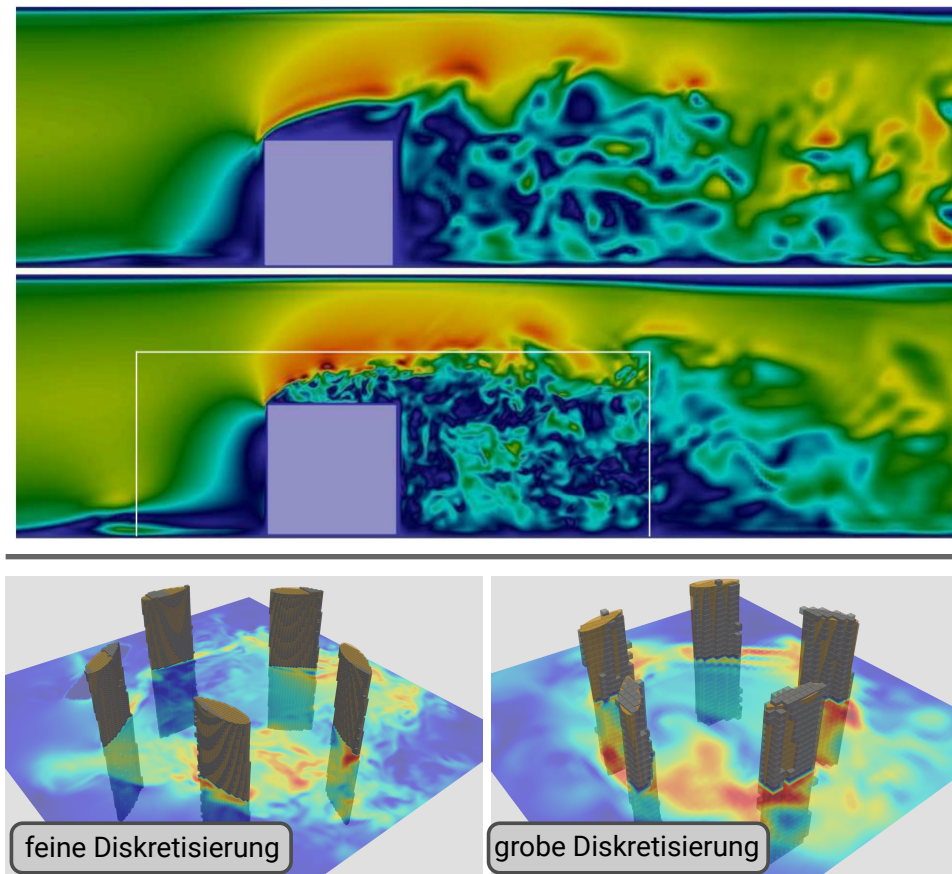


Abbildung 9: Zwei verschiedene Szenarien, die lokale Gitterverfeinerungen des Rechengebietes erforderlich machen bzw. sich im Rahmen der Rechengenauigkeit als vorteilhaft erweisen.

Dieser Abschnitt ist in zwei weitere Unterabschnitte unterteilt, in denen zwei substantiell voneinander unabhängige Gitterverfeinerungstechniken adressiert werden. Während in Unterabschnitt 2.6.1 Erläuterungen zu der Interpolation der LBM-spezifischen Freiwerte (PDF) im Kontext von Einphasenströmungen gegeben werden, wird in Unterabschnitt 2.6.2 auf das algorithmische Sonderproblem der Interpolation der Lage der Wasseroberfläche für Strömungen mit freier Oberfläche eingegangen.

2.6.1 LBM – Verteilungsfunktionen

Für die Simulationen in PROEIS wurde der in Abb. 10 dargestellte Ansatz auf Basis überlappender Gitter gewählt. Die Gittergröße der feinen Vernetzung ist dabei halb so groß wie die Gittergröße der groben Vernetzung ($\Delta x_c = 2\Delta x_f$). Aufgrund der bei LBM-Verfahren vorhandenen starken Koppelung von Gitterweite und Größe des Zeitschrittes führt dies zu einem geschachtelten Zeitschrittverfahren. Daraus folgt, dass ein Zeitschritt im groben Gitter dem Doppelten des Zeitschrittes im feinen Gitter ($\Delta t_c = 2\Delta t_f$) entspricht.

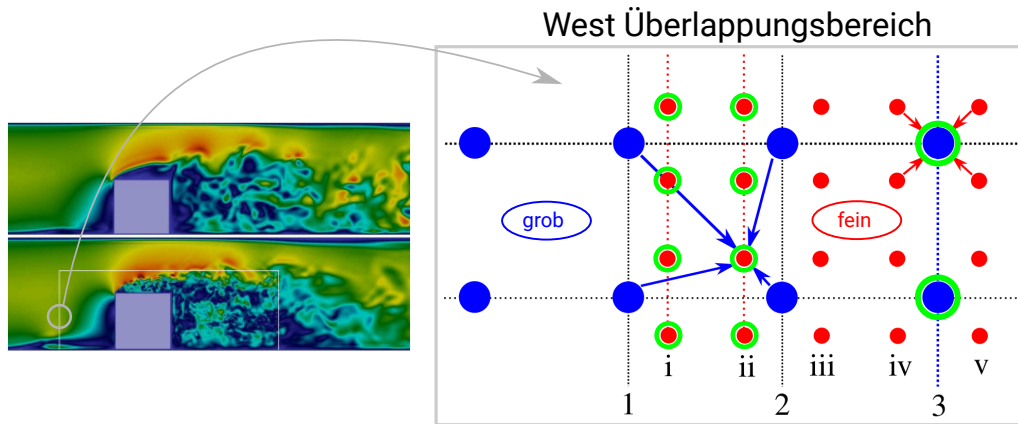


Abbildung 10: Grundlegendes Schemata des bidirektionalen Datenaustausches zwischen zwei Gittern unterschiedlicher Diskretisierung:

Interpolation und Reskalierung der Nichtgleichgewichtsanteile der Verteilungsfunktionen in dem westlichen Überlappungsbereich zwischen zwei Gittern unterschiedlicher Diskretisierung: nach den auf den jeweiligen Gittern ausgeführten Propagationen ungültige Knoten (grün umrandet) müssen durch bi- (2D) bzw. trilineare (3D) Interpolationen valider Verteilungsfunktionen des Hintergrund- (grob \rightarrow fein) respektive Vordergrundgitters (fein \rightarrow grob) als Randbedingungen in den Überlappungszonen aufgeprägt werden.

Nach dem Ende eines abgeschlossenen diskreten Zeitschrittes des groben Gitters – bzw. zwei Zeitschritten des feinen Gitters – gibt es in den Randbereichen der jeweiligen Gitter fehlende Partikelverteilungsfunktionen, die aktualisiert werden müssen. Diese Knoten sind in Abb. 10 grün umrandet. Der Synchronisierungsschritt, der als Randbedingung in den Überlappungszonen der Gitter wirkt, umfasst bei der Gitterverfeinerung eine Interpolation der Partikelverteilungsfunktionen zwischen den Gittern. Die Knoteninformationen auf dem groben Gitter (Schicht **3**) werden durch die einfache Mittelung der vier (in 2D, bzw. acht in 3D) benachbarten Knoten des feinen Gitters aktualisiert. Die Knoteninformationen auf dem feinen Gitter (Schichten **i** und **ii**) werden durch eine im zweidimensionalen Raum bilineare bzw. im dreidimensionalen trilineare Interpolation ermittelt.

Mittels dieses Algorithmus wird ein bidirektionaler Informationsaustausch zwischen den Gittern ermöglicht. Hierbei prägt das grob diskretisierte Fernfeld einerseits das von den entfernt wirkenden Randbedingungen des Rechengebietes abhängende Strömungsfeld auf ein feines Gitter auf, in dem eine detaillierte Abbildung und Analyse der Strömungsgrößen gewünscht ist. Andererseits können Informationen, die im feinen Gitter – bei inaktiven Hintergrundgitter – generierten Informationen auch in das Fernfeld transferiert werden. Zur Vermeidung von Rückkopplungen aufgrund des Informationsaustausches zwischen den Gittern wird ein Abstand in dem Überlappungsbereich zwischen den Interpolationslayern festgelegt, sodass stets valide, nicht-interpolierte PDF in dem jeweiligen Synchronisierungsschritt verwendet werden.

Aufgrund der unterschiedlichen Zeitschrittweiten in beiden Gitterbereichen ist darüber hinaus eine Reskalierung notwendig. Die Gleichgewichtsanteile der Verteilungsfunktionen für das feine und grobe Gitter sind ausschließlich Funktionen der hydrodynamischen, makroskopischen Para-

meter $f^{\text{eq}} = f^{\text{eq}}(\rho, \mathbf{u})$ und sind demnach unabhängig von der gewählten Gitterdiskretisierung. Die unbekanntenen Partikelverteilungsfunktionen werden folglich über Gleichungen bestimmt, in denen eine Reskalierung der Nichtgleichgewichtsanteile erforderlich ist, die via

$$f_{i,\text{grob}} = f_{i,\text{fein}}^{\text{eq}} + r f_{i,\text{fein}}^{\text{neq}} \quad \text{und} \quad f_{i,\text{fein}} = f_{i,\text{grob}}^{\text{eq}} + \frac{1}{r} f_{i,\text{grob}}^{\text{neq}}, \quad (24)$$

erfolgt, wobei $f_{i,\text{grob}}$ bzw. $f_{i,\text{fein}}$ den fehlenden Verteilungsfunktionen im groben bzw. im feinen Gitter entsprechen. In den Beziehungen ist r der Reskalierungsfaktor zwischen den Nichtgleichgewichtsbeiträgen des feinen und groben Gitters, der als

$$r = \frac{\nu_{\text{fein}}(6\nu_{\text{grob}} + 1)}{\nu_{\text{grob}}(6\nu_{\text{fein}} + 1)} \quad (25)$$

definiert ist und die unterschiedlichen Viskositäten auf den beiden Gittern zueinander in Beziehung setzt.

Die Erweiterung des räumlichen Diskretisierungsansatzes wurde zunächst in das Einphasenmodell des Strömungslösers integriert und anhand zahlreicher charakteristischer Testfälle – sowohl laminarer als auch transienter, turbulenter Strömungen – umfangreich und erfolgreich validiert.

2.6.2 VOF – Füllstandsvariable

Zonale Methoden erfordern eine Rekonstruktion bzw. Interpolation von Feldgrößen am Zonenübergang. Während hierfür in Unterabschnitt 2.6.1 anhand von Einphasenströmungen Algorithmen für die Interpolation der Strömungsgrößen (LBM-Partikelverteilungsfunktionen) dargestellt wurden, wird in diesem Unterabschnitt auf eine im Rahmen des Forschungsprojektes entwickelte Methode zur Interpolation der Lage der freien Wasseroberfläche eingegangen.

Die Techniken basieren hierbei auf einer stückweise linearen Rekonstruktion der freien Oberfläche (engl. *Piecewise Linear Interface Reconstruction*, PLIC). Die Implementierung des PLIC-Verfahrens zur Interpolation der Füllstandsvariable der VoF-Methode im Übergangsbereich unterschiedlich diskretisierter Gitter wird in diesem Unterabschnitt für den zweidimensionalen Fall beschrieben.

Die Vorschrift für den Austausch der Informationen des Phasenfeldes – der Füllstandsvariable ϵ der Zellen ($0 \leq \epsilon \leq 1$), welche für die Advektion der freien Oberfläche benötigt wird – basiert auf einer Finite-Differenzen-Approximation des Normalenvektors der Oberfläche. Hierbei wird der Gradient der Füllstandsvariable $\nabla\epsilon$ aus den Informationen der acht umgebenden Nachbarzellen gebildet, vgl. Abb. 11 (Mitte). Aus der anschließenden Normierung ergibt sich der Oberflächennormalenvektor $\mathbf{n} = -\nabla\epsilon/|\nabla\epsilon|$. In der Abb. 11 (rechts) ist eine beispielhafte Zelle mit rekonstruierter Oberfläche gezeigt. Im Vergleich zu der *gravitationsbasierten* Repräsentation (links), die sich durch eine normal zum Gravitationsvektor ausgerichtete Oberfläche auszeichnet, wird die Lage der in rot dargestellten kontinuierlichen freien Oberfläche signifikant besser erfasst.

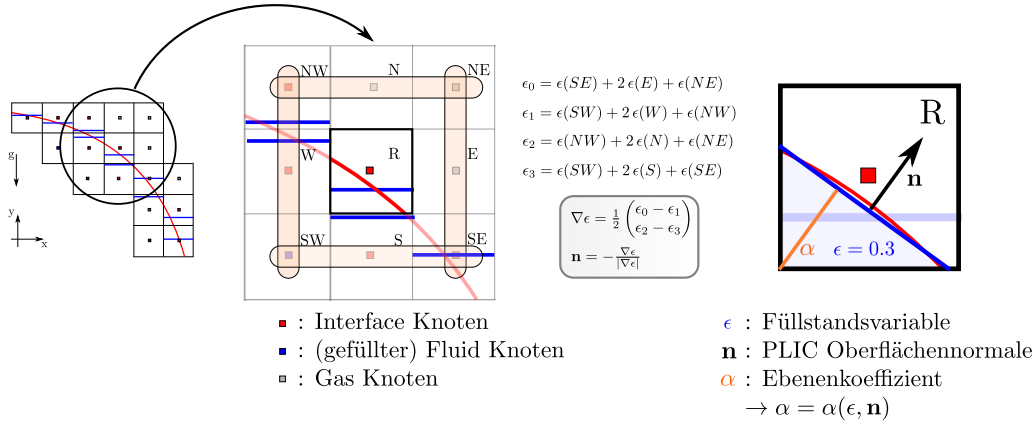


Abbildung 11: PLIC Verfahren - Approximation der Ausrichtung der freien Oberfläche auf Basis der Informationen aus den acht umgebenden Nachbarknoten: *Gravitationsbasiertes* VoF mit Ausrichtung der Oberfläche normal zum Gravitationsvektor (links); Finite-Differenzen-Approximation und Berechnungsvorschrift für den normierten Oberflächenvektor (Mitte); PLIC Notation innerhalb einer Zelle und beispielhafte Ausrichtung der freien Oberfläche (rechts).

Die zusätzlich zum Füllstand der Zellen generierte PLIC Information (PLIC Oberflächennormale und Ebenenkoeffizient $\alpha = \alpha(\epsilon, \mathbf{n})$) wird genutzt, um einen Algorithmus zum Austausch des Phasenfeldes zwischen zwei Gittern unterschiedlicher Diskretisierung zu entwickeln.

Neben der Interpolation der LBM-Verteilungsfunktionen analog zu dem in Abb. 10 dargestellten Schema wird für Strömungen mit freien Oberflächen auf den identischen Schichten in den Überlappungsbereichen des groben bzw. feinen Gitters eine Bestimmung der Füllstandsvariable durchgeführt. Hierzu werden zunächst mittels der linear rekonstruierten Oberfläche jene Knoten bestimmt, die dem Interface ($0 < \epsilon < 1$) zugehörig sind. Mittels der bekannten Information des Ebenenkoeffizienten – $\alpha_{\text{grob}} = \alpha_{\text{grob}}(\epsilon_{\text{grob}}, \mathbf{n}_{\text{grob}})$ – werden anschließend die einzelnen Füllstände der Zellen des feinen Gitters via $\epsilon_{i,\text{fein}} = \epsilon_{i,\text{fein}}(\alpha_{\text{grob}})$ bestimmt. Die Abb. 12 zeigt einen eindeutig definierten Übergang der Füllstandsvariable auf zwei Gittern unterschiedlicher Diskretisierung. Der im Rahmen dieses Projektes entwickelte Gitterverfeinerungsalgorithmus für Strömungen mit freier Oberfläche wurde umfangreich validiert. Die Validierung im zweidimensionalen Raum umfasste dabei zwei Benchmark Testfälle: 1) die Bewegung eines geführten Fluidtropfens ausgeführt als reiner Advektionstest ohne die Lösung der LBM-spezifischen Größen; 2) die Ausbreitung einer durch einen Wellenerzeuger aufgeprägten Soliton-Welle durch ein zonal feiner diskretisiertes Gitter inklusive der Berücksichtigung des LBM Propagations- und Kollisionsschrittes.

Die Umsetzung des zweiten Testfalles ist in Abb. 13 nebst einer kurzen quantitativen Auswertung in Form der Auswertung der L1-Norm des Amplitudenfehlers gezeigt, dessen Absolutwert – bei gleicher Fehlerordnung – unter Verwendung einer Gitterverfeinerungsregion (*Refined Grid*) geringer beziffert ist als im uniformen Fall (*Uniform Grid*). Beide Testfälle konnten die Güte des entwickelten Verfahrens nachweisen. Die Ergebnisse der Validierung wurden darüber hinaus im Mai 2017 auf der *VII International Conference on Computational Methods in Marine Engineering* (MARINE 2017) in Nantes vorgestellt.

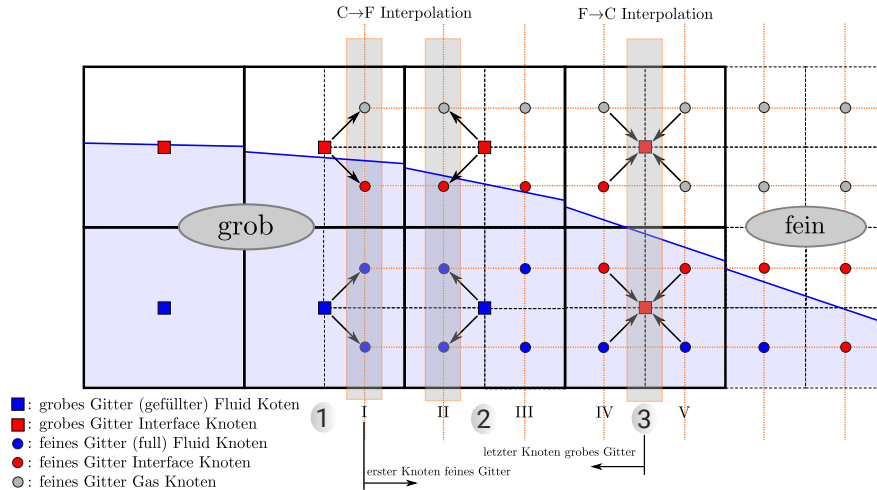


Abbildung 12: Ausrichtung der freien Wasseroberfläche als Detailansicht in dem Überlappungsbereich zwischen zwei Gittern unterschiedlicher Diskretisierung. Die Interface Repräsentation zeichnet sich dadurch aus, dass sie eindeutig ist. D.h. das Interface ist einerseits geschlossen und andererseits an keinem Knoten aufgedickt bzw. doppelt definiert.

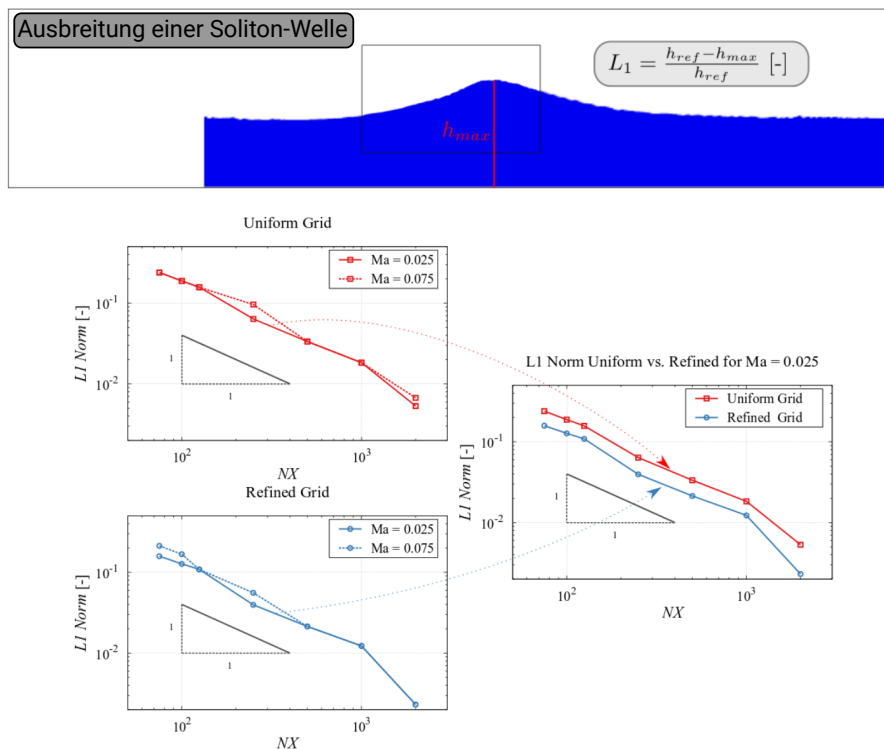


Abbildung 13: Testfall zur Ausbreitung einer durch einen Wellengenerator (bewegtes Geometrieobjekt im linken Teil des Rechengebietes) aufgeprägten Soliton-Welle durch ein Gebiet – als schwarzes Rechteck markiert – mit lokaler Gitterverfeinerung (oben). Quantitative Auswertung des Amplitudenfehlers als L1-Norm der maximalen Wellenerhebung zu einem fixen Zeitpunkt der Simulation (unten). Der Vergleich der L1-Fehlernormverläufe unter Variation der Knotenanzahl zwischen uniformer Diskretisierung und Simulation mit Gitterverfeinerung zeigt den Vorteil aufgrund der feineren Diskretisierung (unten rechts).

Im Rahmen einer umfangreichen Validierung der Methode wurden zusätzliche Korrekturalgorithmen entwickelt und implementiert. Die Algorithmen gewährleisten eine eindeutige Definition (Schärfen der Phasentrennfläche, engl. *Interface Sharpening*) des Übergangsbereiches zwischen der Gas- und simulierten Fluidphase. Diese ist für die Güte des VoF-Verfahrens ausschlaggebend. In Abb. 14 ist das Verfahren zur Behandlung nicht valider – uneindeutig definierter – Interface Konstellationen dargestellt. Sobald im Standardschritt der PLIC VoF Gitterverfeinerung vier (in 2D) Interface Zellen in dem Überlappungsbereich als Interface erkannt werden (links oben), wird zunächst eine gemittelte Grobgitterzelle gebildet (links unten). Anschließend wird der in der Box dargestellte Korrekturalgorithmus zur Bestimmung einer validen Konfiguration durchlaufen.

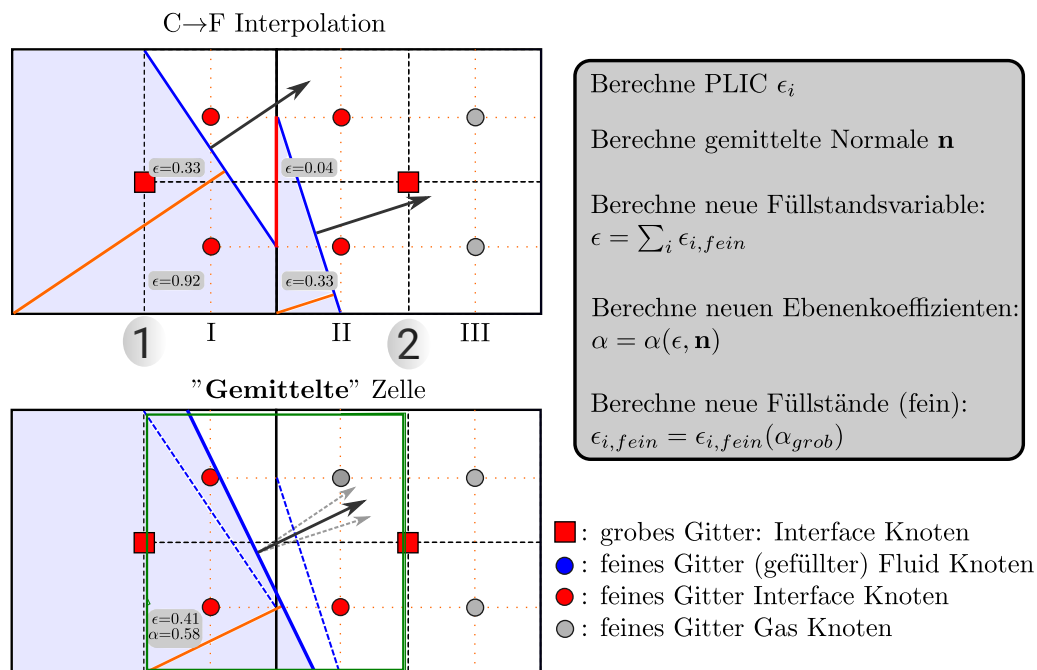


Abbildung 14: Berechnungsvorschrift für die Behandlung nicht eindeutig definierter Interface-Konstellationen. Bildung einer gemittelten Zelle und anschließende Berechnung der Füllstände in den Zellen des feinen Gitters zur Schärfung des Interfaces (engl. *Interface-Sharpning*).

Ein weiterer Schritt, der sowohl die Gitterverfeinerung (LBM Verteilungsfunktionen & VOF Phasenfeld) als auch die korrekte Kraftberechnung in dem Überlappungsbereich einschließt, ist die Behandlung von Starrkörperbewegungen über Gitter variierender Diskretisierungen hinweg. Hierzu wurden neben der Entwicklung der Algorithmen für den Gitterübergangsbereich zudem Modifikationen des Löser hinsichtlich der Integration von Fluidlasten auf den Körper (vgl. Kapitel 5.1.3) sowie der Bewegung bzw. Advektion der Geometrie vorgenommen. Die Kraftintegration sowie Geometrieadvektion erfolgen hierbei jeweils zum Ende eines diskreten Zeitschrittes des groben (Hintergrund-) Gitters.

In der untenstehenden Abb. 15 ist der Testfall zur Untersuchung der Starrkörperadvektion über Gebietsgrenzen hinweg dargestellt. Zur Validierung wird eine zweidimensionale Geometrie als Schnittbene einer beispielhaften generischen dreidimensionalen Eisscholle verwendet (oben

links). Unten links ist der Gitterübergang am Westrand des feinen Gitters als Detail herausgenommen. Das auf den Knoten dargestellte Feld des Geschwindigkeitsbetrages weist hierbei keinerlei Diskontinuitäten auf, woraus zu schließen ist, dass die Interpolation der LBM-Verteilungsfunktionen einwandfrei funktioniert. Zur Nutzung der Standard Interpolationsvorschrift wird hierbei eine Extrapolation der Verteilungsfunktionen auf die in den Überlappungsbereichen auftretenden Geometrienoten ausgeführt. Hierin werden mittels der aus den umgebenden Fluidknoten bestimmten makroskopischen Werte für die Dichte und die Geschwindigkeit Gleichgewichtsverteilungen berechnet und auf den Geometrienoten angesetzt. Die vier Aufnahmen der Serie auf der rechten Seite stellen den Betrag des Geschwindigkeitsfeldes in der Fluidphase dar, während sich die 2D Schollengeometrie aus dem groben Gitter kommend und durch das feine Gitter gehend (schwarz markiertes Rechteck) bewegt.

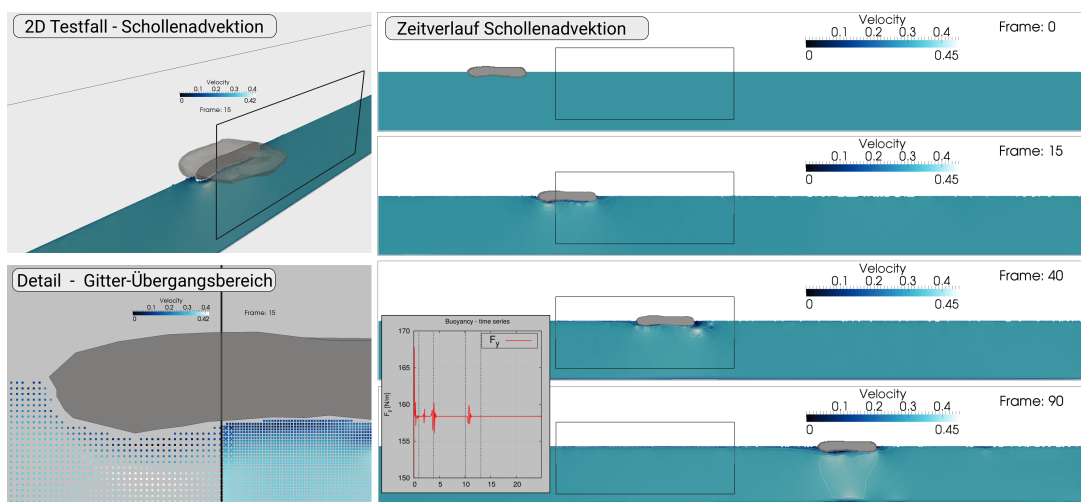


Abbildung 15: 2D Testfall zur Starrkörperadvektion über Gebietsgrenzen variierender Diskretisierung hinweg.

Die Bewegung des Körpers ist hierbei für die Freiheitsgrade in beide Raumrichtungen ermöglicht und die Berechnung erfolgt für den hier untersuchten zweidimensionalen Fall mittels eines auf Quaternionen-basierten Algorithmus.

3 Physik-Bibliotheken

Die Anforderungen an Methoden zur numerischen Simulation der Starrkörperdynamik sind häufig sehr umfangreich und umfassen viele Unterfunktionen. Sie werden daher als eine Physik-Bibliothek (engl. *Physic-Engine*) kompakt zusammengefasst und bieten über Schnittstellen eine portable und unabhängige Möglichkeit zur Anbindung an ein steuerndes Programm.

Die gängige Aufgabe einer Physik-Bibliothek ist die Vorhersage der Körperbewegung unter gegebenen Körperlasten, was auch als Vorwärts-Dynamik bezeichnet wird. Je nach Anwendungszweck unterscheidet sich die erforderliche Exaktheit der simulierten physikalischen Prozesse und ermöglicht eine Klassifizierung der Bibliothek. Im Unterhaltungsbereich ist insbesondere bei Computerspielen eine Echtzeitfähigkeit der Bibliothek erforderlich. Der durch die stark vereinfachten Methoden resultierende Fehler ist visuell unauffällig und damit vertretbar um im Gegenzug eine erhöhte Bildwiedergabe zu erzielen. Hingegen wird für wissenschaftliche Zwecke und in der industriellen Robotik eine möglichst physikalische exakte Simulation angestrebt. Der durch die erhöhte Komplexität auftretende Mehraufwand wird gerne in Kauf genommen. Zur Wahl einer geeigneten Physik-Bibliothek, die im Späteren an den Strömungslöser `elbe` gekoppelt wurde, wurden folgende Anforderungen definiert:

- möglichst exakte und dennoch effiziente Methoden
- quelloffen, plattformunabhängig und eigenständig
- gute Dokumentation (Schnittstellen, interne Methoden) und leichter Einstieg
- ausgewachsen, robust, stabil und in aktiver Pflege
- Kollisionsbehandlung beliebiger Geometrien (insbesondere triangulierte Gitter)
- lizenzkonform zur Anbindung an den Strömungslöser `elbe`

Zu Beginn dieses Forschungsprojekts stand die Nutzung der *pe Physics Engine* (PE) [49, 48] als vielversprechende Physik-Bibliothek in Aussicht. Aufgrund eines zwischenzeitlichen Lizenzwechsels der PE zur GPL¹, ließ sich die Nutzung mit `elbe` nicht weiter vereinbaren. Die GPL setzt ein sogenanntes *starkes Copyleft* ©² voraus, dass die Lizenzhaltung abgeleiteter Werke erfordert. So dürfen insbesondere Programmteile oder -bibliotheken, die unter einer Lizenz mit starkem Copyleft frei zur Verfügung stehen, nur in Programme inkludiert bzw. gelinkt werden, die unter derselben Lizenz stehen. Eine Lizenzierung von `elbe` nach der GPL und der einhergehenden Veröffentlichung, inklusive einer angemessenen Dokumentation, war nicht vereinbar. Die laufende Implementierung und Validierung von PE mit `elbe` musste eingestellt und auf eine andere Bibliothek ausgewichen werden.

¹GNU General Public License version 3: <https://www.gnu.org/copyleft/gpl.html>

²Copyleft ©: <https://www.gnu.org/copyleft/>

Tabelle 3: Verfügbare Physik-Bibliotheken (Stand 2015), Gegenüberstellung der Vereinbarkeit:
■ gut, ■ mäßig, ■ keine

Physik-Bibliothek	Quell-Code Zugang	Lizenz	Verwendungs- zweck	letzte Veröf- fentlichung	Initiali- sierung
<i>AgX Dynamics</i> [8]	kommerziell	proprietär	multi	2014	2007
<i>Bullet Physics Library</i> [25]	frei	Zlib	Unterhaltung	2014	<2006
<i>Havok Physics</i> [46]	kommerziell	proprietär	Unterhaltung	2011	
<i>JigLib</i> [20]	frei	Zlib	Unterhaltung	2007	
<i>JigLibX</i> [2]	frei	MIT	Unterhaltung	2009	
<i>Newton Game Dynamics</i> [56]	frei	Zlib	multi	2014	2003
<i>PhysX</i> , ehemals <i>NovodeX</i> [81]	kommerziell	proprietär	multi	2014	2002
<i>Open Dynamics Engine</i> [89]	frei	LGPL/BSD	multi	2014	2001
<i>OpenTissue</i> [5]	frei	Zlib	multi	2009	2001
<i>PhysicsEngine</i> [49]	frei	GPLv3	multi	2014	2006
<i>Siconos</i> [6]	frei	GPLv2	Forschung	2014	<2007
<i>Tokamak</i> [64]	frei	BSD/Zlib	Unterhaltung	2013	2003
<i>True Axis</i> [7]	frei	?	Unterhaltung	2006	
<i>Vortex Dynamics</i> [22]	kommerziell	proprietär	multi	2014	

3.1 Verfügbare Bibliotheken

Forschungen im Bereich der Robotik sorgten für die Entstehung vieler Bibliotheken mit unterschiedlichen Anforderungen. Meistens sind diese extra auf ihren Aufgabenbereich zugeschnitten und stark optimiert. Als Tailormade-Bibliotheken bieten sie entsprechende Vor- und Nachteile. Ivaldi *et al.* [50] schufen mit der Auswertung ihrer Umfrage einen Überblick über die Präferenzen und Erfahrungen der Benutzer von Physik-Bibliotheken.

Die Tab. 3 bietet einen Überblick über die bekanntesten Physik-Bibliotheken und verdeutlicht zugleich ihre Vereinbarkeit mit den erwähnten Anforderungen. Die Lizenzen BSD³, Zlib⁴ und MIT⁵ zählen zu den freizügigen Lizenzen, die einen quelloffenen Zugang und einen Wechsel der Lizenz erlauben (auch zu einer restriktiveren als der Originallizenz). Die LGPL⁶ ist ebenfalls eine freizügige Lizenz, kommt allerdings mit einem *schwachen Copyleft* daher und setzt zusätzlich den Zugang und die Änderungsmöglichkeit der LGPL-Teile dem Endbenutzer voraus. In der Praxis wird dies durch das linken der unter der LGPL stehenden Bibliothek gewährleistet. Das die Bibliothek nutzende Programm ist somit klar vom LGPL-Werk getrennt und kann unter einer anderen Lizenz stehen (auch proprietären). Die Bibliothek selbst bleibt frei, das heißt quelloffen und modifizierbar, und muss der LGPL weiter gehorchen. Alle in der Tab. 3 aufgezählten Bibliotheken sind plattformunabhängig.

Seugling/Rölin [86] untersuchten den Funktionsumfang von neun Bibliotheken. Sie evaluierten drei (*Open Dynamics Engine*, *NovodeX*, *Newton*) in den Bereichen Reibung, der Drehimpulserhaltung, der Kollisionsauflösung, sowie der Behandlung von Mehrkörpersystemen und verglichen die Berechnungszeiten. In ihren Tests erzielte *NovodeX* fast immer die besten Ergebnis und er-

³Berkeley Software Distribution License: <http://opensource.org/licenses/bsd-license.php>

⁴zlib/libpng License: <http://opensource.org/licenses/Zlib>

⁵Massachusetts Institute of Technology License: <http://opensource.org/licenses/MIT>

⁶GNU Lesser General Public License: <https://www.gnu.org/copyleft/lgpl.html>

langte, weit abgeschlagen, die meisten Bewertungspunkte. Dennoch wählten sie für ihre weitere Arbeit die *Open Dynamics Engine*, da diese frei zur Verfügung steht und ihren Anforderungen am Nächsten kam.

Boeing/Bräunl [17] stellten einen Vergleich zu Funktionsumfang und Genauigkeit von sieben Bibliotheken (*PhysX*, *Bullet*, *JigLib*, *Newton*, *Open Dynamics Engine*, *Tokamak*, *True Axis*) auf. Im Fokus der Entwicklung von Computerspielen evaluierten sie Integrationsmethoden, Material- und Reibungsmodelle, sowie Kopplungs- und Kollisionsmethoden. Sie beschränkten sich bei ihrer Untersuchung auf primitive Geometrien und stellten fest, dass jede der untersuchten Bibliothek ihre Vorzüge und Schwierigkeiten hat. Dennoch betonten sie die Bequemlichkeit der Verwendung von PAL, um schnell verschiedene Bibliotheken zu testen. Der *Physics Abstraction Layer* (PAL) ist ein sogenannter Bibliotheken-Wrapper und dient als universelle Schnittstelle zwischen Anwendung und Bibliothek. Wrapper wie PAL ermöglichen einen schnellen und einfachen Wechsel der Bibliotheken und erfordern kaum bis gar keine Kenntnis über die Bibliotheken selbst. Weitere bekannte Wrapper sind *Open Physics Abstraction Layer* (OPAL) und *GangstaWrapper*.

Im Fokus der Roboter montagesimulation in der virtuellen Realität stießen Hummel *et al.* [47] mit *Bullet* an die Grenzen der Bibliothek. Sie testeten *Bullet* gegen vier andere Bibliotheken (*Newton*, *Havok*, *Open Dynamics Engine*, *PhysX*). Hierzu untersuchten sie die Geschwindigkeit der Kollisionserkennung, die Genauigkeit der Kollisionsantwort, die Überlappungstiefe bei Kollisionen, sowie die Behandlung von Mehrkörpersystemen und simulierten die komplexe Herstellung einer Schraubverbindung mit Hilfe von konvexen Geometrien. Keine Bibliothek konnte in allen Versuchen als Sieger hervorgehen. Sie kamen daher zu einem ähnlichen Schluss wie Boeing und Bräunl [17], dass keine Bibliothek die Ideale sei. Dennoch erwägen sie weitere Untersuchungen an *Newton* und *PhysX* zu führen.

Metrikin *et al.* [74, 71] evaluierten vier Physik-Bibliotheken (*AgX Dynamics*, *Open Dynamics Engine*, *PhysX*, *Vortex*) um eine im Weiteren zur Simulation von Eis-Fluid-Struktur Interaktionen zu nutzen. Sie untersuchten die Integrationsmethode, die Kollisionserkennung, sowie den Dokumentationsumfang und kamen ebenfalls zum Schluss, dass jede Bibliothek Stärken und Schwächen habe und keine perfekt sei. Außerdem stellten sie fest, dass sich kommerzielle Bibliotheken meistens durch eine ausgezeichnete Dokumentation und Unterstützung der Entwickler kennzeichnen und sich eine Investitionen unter Umständen lohnen kann.

Mit Hilfe der Tab. 3 ließ sich die Menge der Bibliotheken, die den Anforderungen genügen, auf die *Open Dynamics Engine* und *Newton* reduzieren. Nach Seugling/Rölin [86] wies die *Open Dynamics Engine* eine der ausführlichsten Dokumentationen auf. Außerdem bot *Newton* keine direkte Unterstützung für die Kollisionsbehandlung triangulierter Gittergeometrien. Die Wahl einer Physik-Bibliothek, die im Weiteren Verwendung findet, fiel somit auf die *Open Dynamics Engine*.

Mit dem Ziel, die volle Kontrolle über die Bibliothek und Kenntnis über die internen Methoden zu erlangen, wurde die Nutzung eines Bibliothek-Wrappers nicht in Erwägung gezogen.

4 Open Dynamics Engine

Die *Open Dynamics Engine* (ODE) [4, 89] ist eine quelloffene und plattformunabhängige C/C++ Physik-Bibliothek. Sie dient zur Simulation der Dynamik von kollidierenden Starrkörpern und ist wahlweise unter der BSD oder der LGPL lizenziert.

ODE zählt zu den bekanntesten Physik-Bibliotheken. Sie gilt als schnell, robust, stabil und ausgereift, allerdings auch als verwaist [50]. Die Engine findet in einigen Computerspielen, vielen Robotik Projekten und Open World Simulatoren, sowie in kommerziellen Programmen Anwendung. Ihr Autor Russell Smith betonte, dass die Bibliothek sich eher durch ihre Stabilität und Geschwindigkeit als ihre Genauigkeit kennzeichnet [89].

Im Folgenden wird die Funktionsweise ODEs in der Version v0.13 vom Februar 2014 präsentiert und die zugrundeliegenden Theorien und Modelle erläutert.

Die ODE kennzeichnet sich durch folgenden **Funktionsumfang**:

- Starrkörper mit beliebiger Massenverteilung.
- Bewegungsgleichungen werden im Impulsbereich gelöst.
- Integrierte Kollisionserkennungsmethode und zwei Reibungsmodelle.
- Unterstützung gängiger Geometrien (primitiv, konvex und trianguliert).
- Beschleunigte Kollisionserfassung durch Hüllkörper, Hierarchien und Gruppierungen.
- Verwendung eines harten Kontaktmodells.
- Unterstützung von Mehrkörpersystemen durch bilaterale Kopplungen.
- Dynamische Kopplungen wie Federn, Dämpfer, Drehmomenten- und Linearmotoren.
- Zwei Lösungsmethoden (direkt und iterativ).
- Semi-implizites Zeitschrittverfahren erster Ordnung.

Die **Dokumentation** ODEs fällt für wissenschaftliche Zwecke sehr dürftig aus. Dies mag daran liegen, dass ODE mit der Zeit zu einem offenen Gemeinschaftsprojekt wurde und die Dokumentationen einen geringeren Stellenwert bezog. Neben einem in die Jahre gekommenen Handbuch [89], das sich an Einsteiger richtet und grundlegende Schnittstellen der Bibliothek beschreibt, existiert ein etwas ausführlicheres Wiki [4]. Dennoch lässt sich in beiden Quellen nur schwer bis gar nicht nachlesen, was hinter den Schnittstellen passiert und auf welche wissenschaftlichen Methoden die Bibliothek zurückgreift. Selbst die wenigen Kommentare im Quellcode fallen nichtssagend aus. Nur durch die Verwendung von Internet-Suchmaschinen und Artikeln Dritter lässt sich dies zufällig herausfinden. Für Fragen und Diskussionen steht seit April 2007 eine Mailing-Liste auf Google-Groups [1] zur Verfügung. Frühere Beiträge lassen sich in einem Archiv [3] auffinden.

ODE kommt mit einer Sammlung kleiner demonstrativer Programme daher, die den grundlegenden Funktionsumfang präsentieren sollen. Leider sind auch diese nur spärlich dokumentiert und wurden mit der fortschreitenden Entwicklung ODEs kaum weiter gepflegt.

Wie bereits erwähnt, finden sich **Evaluationen** in [86, 17, 47, 71, 61]. Alle haben gemein, dass sie mit einer älteren Version von ODE geführt wurden und Kollisionen mit triangulierten Gittergeometrien außer Acht ließen.

In mehreren Projekten wurde ODE zur Steigerung der Genauigkeit und Effizienz modifiziert. Durch Parallelisierungen in der Kollisionserfassung und -behandlung, sowie Optimierungen am Speichermanagement ODEs erzielten Sreeram/Pande [92] eine Beschleunigung von $\approx 27\%$. Anzumerken ist, dass dieses Ergebnis in einem nur schwer parallelisierbaren Szenario erreicht wurde. Vergleichbare Ergebnisse erlangten Reinders [84] und Ashley-Rollman [11], die ähnliche Modifikationen durchführten. ODE weist in der aktuellen Version Parallelisierungen in den beschriebenen Bereichen auf, allerdings lässt sich nicht feststellen aus welchem der genannten Projekte diese entstammen.

Zur Anwendung in der Robotik nahmen Drumwright *et al.* [31] Untersuchungen an ODE vor. Sie erweiterten ODE um viskose Gelenke, führten ebenfalls Parallelisierungen ein und optimierten die Kollisionsmethode für konvexe Geometrien. Ihre Arbeit scheint allerdings nicht in den offiziellen ODE-Quellcode eingeflossen zu sein.

Technisch gesehen ist ODE nahezu komplett in C++ geschrieben, nutzt Methoden der Objektorientierten Programmierung und bietet Schnittstellen in C. Außerdem gibt es die Möglichkeit zur Anbindung an die Skriptsprache Python. Die Bibliothek spaltet sich strikt in zwei Teile auf: der Starrkörperdynamik und der Kollisionsoperation. Diese durchaus sinnvolle Auftrennung von Körpern (Zustand und Eigenschaft) und ihren Geometrien (Ausdehnung) sorgt für eine übersichtlichere Gestaltung. Körpergeometrien sind somit nur für die Kollisionsmethode relevant und Bewegungs- und Trägheitseigenschaften nur für die Lösung der Bewegungsgleichung (inklusive der Kollisionsantwort).

4.1 Kollisionsmethode

Um Geometrieüberlappungen während der dynamischen Starrkörpersimulation zu vermeiden, befasst sich die Kollisionsmethode mit der Erfassung möglicher Geometriekollisionen und generiert entsprechende Kollisionsantworten. Das Auffinden von etwaigen Kontaktregionen gilt als nichtlineares geometrisches Problem und kann sich sehr aufwendig gestalten [33, 78]. ODE, wie auch viele andere Bibliotheken, diskretisiert aus Effizienzgründen die Kontaktflächen als eine Menge von Kontaktpunkten.

Folgend wird erläutert, wie ODE Kollisionen in zwei Schritten – der Kollisionserkennung und der Kollisionsbehandlung – berücksichtigt.

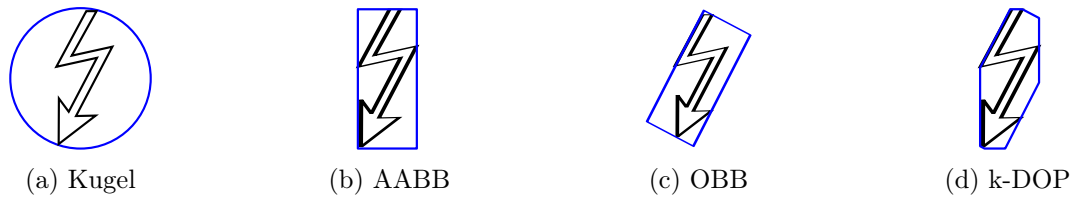


Abbildung 16: Gängige Hüllkörperformen, vereinfacht im Zweidimensionalen dargestellt.

4.1.1 Kollisionserkennung

Die Kollisionserkennung untersucht, ob Körpergeometrien sich überlagern und berechnet entsprechende Kontaktpunkte.

ODEs Kollisionserfassungsmethoden sind teilweise in einer weiteren Kollisionsbibliothek ausgelagert. Insbesondere bei der Verwendung von triangulierten Gittergeometrien greift ODE auf eine externe Bibliothek zurück. Die Kollisionsbibliotheken, OPCODE [98] – die in der Standardkonfiguration Verwendung findet – und GIMPACT [66], liegen dem Quellcode von ODE bei und lassen sich bei der Kompilierung von ODE umstellen. Falls eine andere Kollisionsbibliothek gewünscht ist, muss diese nur den Typ- und Namenskonventionen von ODE entsprechen.

Mit ODE lassen sich alle gängigen **Geometrien** miteinander kollidieren. Neben Primitiven – wie Zylinder, Kugel, Kapsel, Quader und Ebene – werden konvexe und triangulierte Gittergeometrien, sowie Höhengitter unterstützt. Außerdem kann der Strahl als ein Hilfsmittel eingesetzt werden.

Die Kollisionserfassung ist in der dynamischen Starrkörpersimulation im Allgemeinen der zeit-aufwendigste Teil mit $\mathcal{O}(n^2)$, da alle Geometrien gegeneinander untersucht werden müssen. Mit bestimmten Techniken und Ausschlussverfahren lassen sich Kollisionen zwischen entfernten Geometrien schnell disqualifizieren und viel Berechnungszeit sparen.

Hüllkörper dienen in der Kollisionserfassung als ein schnelles Mittel zum Ausschließen von Geometrieüberschneidungen. Sie sollen auf möglichst einfache und effiziente Weise ein minimales räumliches Volumen beschreiben, in dem die Geometrie gerade noch Platz findet. Die simpelste Form eines Hüllkörpers ist die eines achsparallelen Quaders (engl. *Axis Aligned Bounding Box*, AABB). Neben AABB gibt es viele weitere Hüllkörperformen wie zum Beispiel die Kugel, den gerichtete Quader (engl. *Oriented Bounding Box*, OBB) oder das konvexe Polyeder (engl. *k-Discrete Oriented Polytopes*, k-DOP). In Abb. 16 sind diese Hüllkörper zusammengefasst. Sie unterscheiden sich in Hülleffizienz, Speicherbedarf und -zugänglichkeit, sowie Erst- und Neuberechnungsaufwand. ODE unterstützt nur die AABB als Hüllkörper.

Optional bietet ODE die Möglichkeit, Geometrien in sogenannte **Spaces** zu gruppieren. Dies erfolgt manuell und erlaubt es in einer *Space* auch andere *Spaces* und Geometrien hierarchisch zu organisieren. *Spaces* unterliegen einem priorisierten Kollisionstest. Insbesondere wenn viele Körper an mehreren entfernten Stellen gruppiert sind und jeweils einem *Space* angehören, können sie für eine beschleunigte Kollisionserfassung sorgen. Genau wie bei Geometrien, werden auch für die *Spaces* Hüllkörper generiert.

ODEs Kollisionserfassung ist, wie auch bei vielen anderen Physik-Bibliotheken, zweiphasig gestaltet. Die beiden aufeinander aufbauende Phasen, die Fern- und die Nahphase, werden im Weiteren erläutert.

Fernphase Die Fernphase (engl. *broad-phase*) sucht nach potentiellen Geometrieüberschneidungen und testet Überlappungen von Hüllkörpern. Da Hüllkörper den Testaufwand nur konstant reduzieren⁷, werden Hierarchien von Hüllkörpern zur weiteren Beschleunigung verwendet. ODE unterstützt vier Typen von **Hüllkörperhierarchien**:

- **SimpleSpace** ist lediglich eine Liste, die alle Geometrien gegeneinander testet und sich sehr gut eignet, wenn nur wenige Geometrien vorliegen. Der Aufwand bleibt unverändert $\mathcal{O}(n^2)$.
- **QuadTreeSpace** arbeitet raumunterteilend. Sie nutzt eine viergliedrige Baumstruktur, in der geviertelte Unterräume als Kindknoten abgespeichert werden. Dies wird so oft wiederholt, bis der nächste Unterraum kleiner als die AABB der kleinsten Geometrie ist, die sich im Unterraum befindet, oder ein Abbruchkriterium erfüllt wird. In ODE können Kriterien die Tiefe des Baumes, die minimale Größe des Unterraums oder die Anzahl der beinhaltenden Geometrien sein. Außerdem bietet ODE die Möglichkeit **QuadTreeSpace** auf einen bestimmten Bereich anzuwenden. Überlappungstests beginnen im äußersten Geäst und laufen im Notfall bis zur Wurzel hinauf. Dieser Typ eignet sich besonders gut, wenn die Geometrien gleichmäßig verteilt sind und sich nicht auf einem Haufen befinden.
- **HashSpace** scheint eine sortierte Hashtabelle der gesamten **QuadTreeSpace**-Baumstruktur zu erstellen. Tests lassen sich anschließend sehr schnell durchführen. Ausführlicheres lässt sich in der Dokumentation leider nicht finden.
- **SAPSpace** wird erst seit kurzem von ODE unterstützt und basiert auf dem *Sweep and prune* Algorithmus von Terdiman [97]. Er projiziert die Grenzen der Hüllkörper auf die Weltachsen und generiert daraus eine sortierte Liste. Findet sich dabei ein Anfang oder Ende eines Hüllkörpers zwischen Anfang und Ende eines anderen und das auf allen Achsen, so liegt eine potentielle Überschneidung vor. Das besondere an **SAPSpace** ist, dass die Liste im nächsten Zeitschritt weiter verwendet wird und der Algorithmus insbesondere dann beschleunigend wirkt, wenn die Geometrien nur eine geringe Relativbewegung aufweisen.

Sobald das Ausschlussverfahren abgeschlossen ist und eine Liste mit potenziell überlappenden Geometriepaaren generiert ist, wird diese an die zweite Phase, die Nahphase, weitergeleitet.

Nahphase Die Nahphase (engl. *narrow-phase*) prüft nach Überschneidungen zwischen den eigentlichen Geometrien. Speziell auf die Geometriepaare zugeschnittene Algorithmen (etwa

⁷unter der Annahme, dass keine Geometriegruppierung durch *Spaces* erfolgt, gilt weiterhin $\mathcal{O}(n^2)$, nur für eine rechnerisch einfachere Methode

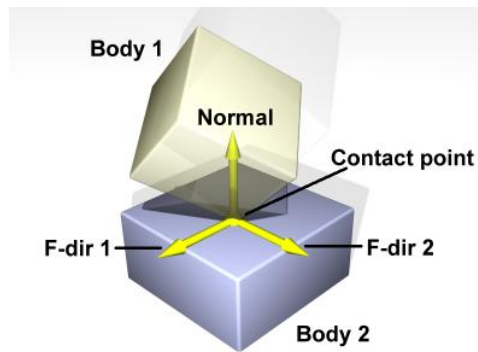


Abbildung 17: Kollision zweier Körper. Kontaktgelenk, bestehend aus Kontaktpunkt, erster und zweiter Kollisionsrichtung und Kollisionsnormale [4].

Cylinder-vs-Kugel oder Trimesh-vs-Trimesh) testen, ob eine echte Überlagerung der beiden potenziell kollidierenden Geometrien vorliegt.

Wichtig ist, dass diese Algorithmen nicht nur ein *ja/nein* als Resultat liefern, sondern bei etwaigen Kollisionen auch Kollisionspunkte berechnen, denen Kollisionstiefe und -ebene zugeordnet werden. Die Kollisionsebene spannt sich aus erster und zweiter Kollisionsrichtung auf und definiert die Kollisionsnormale, vgl. Abb. 17. Im Falle von triangulierten Gittergeometrien greift ODE zur Berechnung der Kollisionspunkte, inklusive ihrer Eigenschaften, auf die zur Kompilierungszeit gewählte Kollisionsbibliothek zurück. Anzumerken ist, dass für eine detektierte Geometrieüberlappung meistens mehrere Kollisionspunkte generiert werden. Anschließend werden diese in einer minimierten Menge zusammengefasst, die den realen Kontakt ausreichend approximativ repräsentieren sollen. Beispielsweise würde eine Fläche-vs-Fläche-Kollision zweier Quader vier Kollisionspunkte erzeugen und wäre damit überbestimmt. Für die Beschreibung der Situation sind allerdings drei ausreichend [102].

In der Praxis zeigt das Kollisionsdetektionsverfahren eine ausreichend gute Effizienz, selbst für viele hoch aufgelöste Geometrien. Eine Implementierung von Kd-Bäumen zur Beschleunigung des Verfahrens wurde daher nicht in Erwägung gezogen.

4.1.2 Kollisionsbehandlung

Nachdem Kollisionspunkte gefunden und für jeden Kollisionstiefe und -ebene berechnet wurden, kann die Kollisionsbehandlung durch die Generierung einer Kollisionsantwort erfolgen.

ODE verwendet ein sogenanntes **hartes Kontaktmodell** [41]. Dieses beschreibt die Stoßmechanik im Impulsbereich und erlaubt die Steuerung der Stoß-Elastizität über den Restitutionskoeffizienten bzw. die Stoßzahl k . Kollisionspunkte werden mittels LAGRANGE-Multiplikatoren als unilaterale Kontaktgelenke temporär für den aktuellen Zeitschritt in das Bewegungsgleichungssystem integriert. Die Körper werden somit, vergleichbar mit Gelenken eines Mehrkörpersystems, in ihrer Bewegung relativ zueinander eingeschränkt. Im Kollisionszeitschritt sorgt der Impulsaustausch für eine sofortige Umkehrung der Körpergeschwindigkeit und lässt die kollidierenden Körper somit nur *einen Zeitschritt weit* überlappen. Diese sofortige Geschwindigkeitsumkehrung

stellt allerdings einen un stetigen Eingriff in das numerische System dar [108, 93]. Die Kollision realer Körper kann somit nur als ein Ereignis erfasst werden, dass räumlich diskret durch Kollisionspunkte beschrieben wird und zwischen zweier Zeitschritte stattfindet.

Einige andere Bibliotheken nutzen ein weiches Kontaktmodell, das aus virtuellen Federn und Dämpfer besteht, die die Relativgeschwindigkeit über weit mehr Zeitschritte stetig umkehren und sehr fehleranfällig sind, so Güttler [44].

Wichtig ist, dass ODE Kollisionen nicht im Voraus erahnen kann. Zur Generierung einer Kollisionsantwort muss immer eine Geometrieüberlappung vorliegen, die aus dem vorherigen Zeitschritt hervorgeht.

Reibungsmodell ODE stellt zwei Reibungsmodelle zur Auswahl. Beide sind fest im Kollisionsmodell integriert und basieren auf dem Reibungsgesetz von COULOMB

$$|f_T| \leq \mu |f_N|, \quad (26)$$

wobei f_T und f_N für die tangentielle Reibungskraft respektive Normalkraft und μ für den Reibungskoeffizienten stehen. Beide Modelle nehmen aus Effizienzgründen und einer einfacheren Implementierung eine starke Approximation des Reibungskegels in Kauf:

- Modell №1: Der Reibungskegel wird durch eine Reibungspyramide approximiert, deren Kanten parallel zur ersten und zweiten Hauptachsenrichtung ausgerichtet sind [101]. Das Reibungsmodell wird somit anisotropie aufweisen, wie es Seugling/Rölin [86] mit einer früheren Version von ODE demonstrierten, die je nach Reibrichtung die Reibkraft um den Faktor $\sqrt{2}$ unterschätzen kann. Im Weiteren wird die Normalkraft unter der Annahme berechnet, dass zunächst keine Reibung vorliegt. Anschließend wird die maximale Tangentialkraft $f_{\max} = \mu |f_N|$ berechnet. Das Modell ist eine Vereinfachung des von Stewart/Trinkle [95, 94] präsentierten Reibungsmodells, dass den Reibungskegel mittels Polygonen beliebig fein annähert.
- Modell №2: Der Parameter μ wird als maximale tangentielle Reibungskraft aufgefasst, die an einem Kontaktpunkt in Hauptachsenrichtung tangential auftreten kann. Dazu wird $f_N = 1$ gesetzt [4]. Dieses Modell ist zwar am einfachsten zu berechnen, jedoch physikalisch äußerst inkorrekt, da die Normalkraft komplett vernachlässigt wird.

Es ist eindeutig, dass das Modell №1 dem Modell №2 vorzuziehen ist, auch wenn dieses nur mäßig genau ist. Da die Reibungsmodelle eng im Lösungsverfahren integriert sind würde eine Überarbeitung der Reibungsmodelle eine äußerst umfangreiche Einarbeitung in den Quellcode der Physik-Bibliothek erfordern, die nicht vereinbar ist.

ODE bietet keine Möglichkeit zur Simulation der dynamischen Reibung. Liegt eine Relativgeschwindigkeit zwischen zwei in Kontakt stehenden Körpern vor, wird weiterhin die approximative statische Reibung angewandt. Rollreibung lässt sich mit ODE um alle drei Achsen simulieren. Wie diese genau behandelt wird, ist leider nicht dokumentiert.

Kollisionsantwort Die Kollisionsantwort lässt sich in ODE individuell und umfangreich über 14 Parameter konfigurieren: Stoßzahl bzw. Restitutionskoeffizient k , Reibungskoeffizient der ersten und zweiten Reibungsrichtung $\mu_{1,2}$, Rollreibungskoeffizient (dreiaxsig) $\mu_{r1,2,3}$, lokale Korrekturzahlen und weitere spezielle Parameter, sowie Schalter zum deaktivieren ganzer Funktionen oder zur Gleichbehandlung der Reibungsachsen. Dem Benutzer ist es auch erlaubt, die Kollisionsantwort ganz zu überspringen und die Geometrien durchdringen zu lassen. Der Möglichkeiten scheinen bisher keine Grenzen gesetzt. Denkbar wäre auch das verschmelzen zweier Körper zu einem, inklusive Trägheits- und Energieerhaltung. Allerdings kann die Schar an Konfigurationsparametern es dem Benutzer auch erschweren das gewünschte physikalische Verhalten zu erzielen.

Nach Konfiguration der Kollisionsantwort wird das Kontaktgelenk einer Kontaktgruppe hinzugefügt, die eine einfachere Organisation aller Kontakte ermöglicht. Anschließend werden diese dem System hinzugefügt und im bevorstehenden Lösungsschritt mit berücksichtigt.

4.2 Lösungsverfahren

Nachdem Kollisionspunkte gefunden und konfiguriert sind, folgt die Lösung des Impuls-basierenden Bewegungsgleichungssystems als ein lineares Komplementaritätsproblem (engl. *linear complementary problem*, LCP [23]). Während geometrische Gelenke mit Hilfe von bilateralen Kopplungsgleichungen modelliert werden, erfolgt die Beschreibung von Kollisions- bzw. Kontaktpunkten mit Hilfe eines harten Kontaktmodells [41] durch unilaterale Kopplungsgleichungen, die dem LCP temporär hinzugefügt werden. Da Kollisionen nur zwischen nicht-verformbaren Starrkörpern stattfinden, lassen sich die Bewegungsgleichungen auf ein Impuls-basiertes Bewegungsgleichungssystem reduzieren. Dies wird im Folgenden gezeigt.

Zunächst gilt die Newton-Euler Gleichung $\mathbf{M} \mathbf{a} = \mathbf{F}_E + \mathbf{F}_C$, wobei \mathbf{F}_E die externen Kräfte und \mathbf{F}_C die Kopplungskräfte beschreiben. Für Letztere gilt $\mathbf{F}_C = \mathbf{J}^T \boldsymbol{\lambda}$ mit \mathbf{J} als die Jacobi-Matrix der Kopplungsgleichungen sowie den unbekanntem Lagrange-Multiplikatoren $\boldsymbol{\lambda}$, vgl. Baraff [14] und Erleben [33]. Kopplungen von Starrkörpern seien durch die kinematische Zwangsbedingung $\mathbf{J} \mathbf{v} = \mathbf{c}$ beschrieben, wobei \mathbf{c} die zeitliche Ableitung der Kopplungsgleichung und \mathbf{v} die Geschwindigkeit der Körper kennzeichnen. Anschließend lässt sich mit einer Upwind-Finite-Differenzen Diskretisierung der Beschleunigung \mathbf{a}^{t+1} und Umordnung das LCP aufstellen

$$\left[\mathbf{J}^t \mathbf{M}^{-1} \mathbf{J}^{tT} \right] \boldsymbol{\lambda}^{t+1} = \frac{\mathbf{c}^{t+1}}{\Delta t} - \mathbf{J}^t \left(\frac{\mathbf{v}^t}{\Delta t} + \mathbf{M}^{-1} \mathbf{F}_E^t \right), \quad (27)$$

wobei $\lambda_i \geq 0$ und $c_i \geq 0$ für unilaterale Kopplungen gelten, vgl. Anitescu und Potra [9] sowie Drumwright *et al.* [31]. Die Gleichung 27 besitzt die Form $\mathbf{A} \boldsymbol{\lambda} = \tilde{\mathbf{c}} + \mathbf{b}$, wobei $\boldsymbol{\lambda}$ und $\tilde{\mathbf{c}}$ komplementär zueinander sind. ODE bietet zwei Methoden zum Lösen des LCP an: eine numerisch exakte und eine iterative Methode. Erstere basiert auf einer optimierten Form des Danzig Algorithmus [24] und findet in der vorliegenden Arbeit Anwendung. Die iterative Lösungsmethode nutzt eine sukzessive Überrelaxation (engl. *successive overrelaxation*, SOR) der Punkt-Gauss-

Seidel Methode (PGS) und eignet sich für sehr große Systeme, wie Arechavalety *et al.* [10] und Drumwright *et al.* [31] zeigen. Mit $\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t \mathbf{v}^{t+1}$ weist die gesamte Lösungsmethode einen semi-impliziten Charakter auf [95, 91].

ODE weist zwar Implementierungen für variable Zeitschrittverfahren auf, von einem Gebrauch dieser wird allerdings abgeraten [4]. Bei Variation der Zeitschrittweite wird insbesondere die Kollisionsmethode Probleme aufweisen. Beispielsweise wird ein vermeintlich ruhender Körper in der weiteren Simulation unterschiedlich tief eintauchen und im Grenzfall anfangen stark zu hüpfen (auch bekannt als *jittering*). Erleben [33] begründet den Gebrauch einer konstanten Zeitschrittweite darin, dass nur ein LCP pro Zeitschritt gelöst werden muss.

Korrekturparameter Da Gleichung 27 die Kopplungszwangsbedingungen im Zustandsraum vernachlässigt, stellt ODE einen ERP-Korrekturparameter (engl. *error reduction parameter*) zur Verfügung, der auf dem Penalty-Prinzip basiert. Des Weiteren kann mit dem CFM-Parameter (engl. *constraint force mixing*) durch dominieren der Hauptdiagonalen der Systemmatrix ein weiches Verhalten der Kopplungen und damit ein stabileres Lösungsverfahren erzielt werden. Näheres zu ERP und CFM findet sich in [33].

4.3 Implementierung

Die ODE wurde in ihrer jüngsten Version v0.13 vom Februar 2014 unmodifiziert und in doppelter Genauigkeit als statische Bibliothek, inklusive der Kollisionsbibliothek OPCODE, folgendermaßen installiert:

```
$ cd ode-*
$ ./configure --enable-double-precision --disable-demos --with-drawstuff=none
$ make -j
# make install
```

Die CMake-Projektdateien von **elbe** wurden um Makros zum Auffinden der Header- und Bibliotheksverzeichnisse erweitert, erlauben das komplette Deaktivieren von ODE und konfigurieren den Compiler automatisch mittels Makefiles. Die Abb. 18 beschreibt den grundlegenden Ablauf einer **elbe**-Simulation in Kombination mit ODE.

Für die Implementierung mussten einige Teile von **elbe** umgeschrieben und erweitert werden. Insbesondere war erforderlich, dass **elbe** nun im Stande ist, in allen Bereichen (Fluidlasten, Geometrieverschiebung, *etc.*) zwischen mehreren Körpern zu unterscheiden. Zuvor ließ sich die Dynamik von nur einem Körper simulieren. Die den Körper betreffenden Informationen (Masse, Geschwindigkeit, *etc.*) waren stark fragmentiert und werden nun für jeden Körper in einer Instanz der Klasse **MCBody3D** zusammengefasst.

Die Klasse **LBM3DInteractor** dient als Schnittstelle zwischen der Fluid- und der Starrkörperdynamik. Sie bietet Speicherplatz für die **elbe**-Geometrien und der Steuereinheit – dem Starrkörperlöser. Von dieser Steuereinheit, der Klasse **MotionCalculator3D**, lassen sich unterschiedliche Bewegungslöser ableiten und anpassen. Für ODE ist dies die Klasse **MCOpenDynamicsEngine**. Sie

dient als *Wrapper* zwischen der ODE-Bibliothek und `eibe` und beinhaltet viele weitere Werkzeuge um mit der Bibliothek zu kommunizieren. Außerdem enthält sie den entscheidenden Teil der Kollisionsmethode, die `nearCallback`-Funktion. Sie steuert die Nahphase der Kollisionserkennung auf eine rekursive Weise, berücksichtigt *Space*-Hierarchien und dient zur Generierung und Konfiguration der Kontaktpunkte.

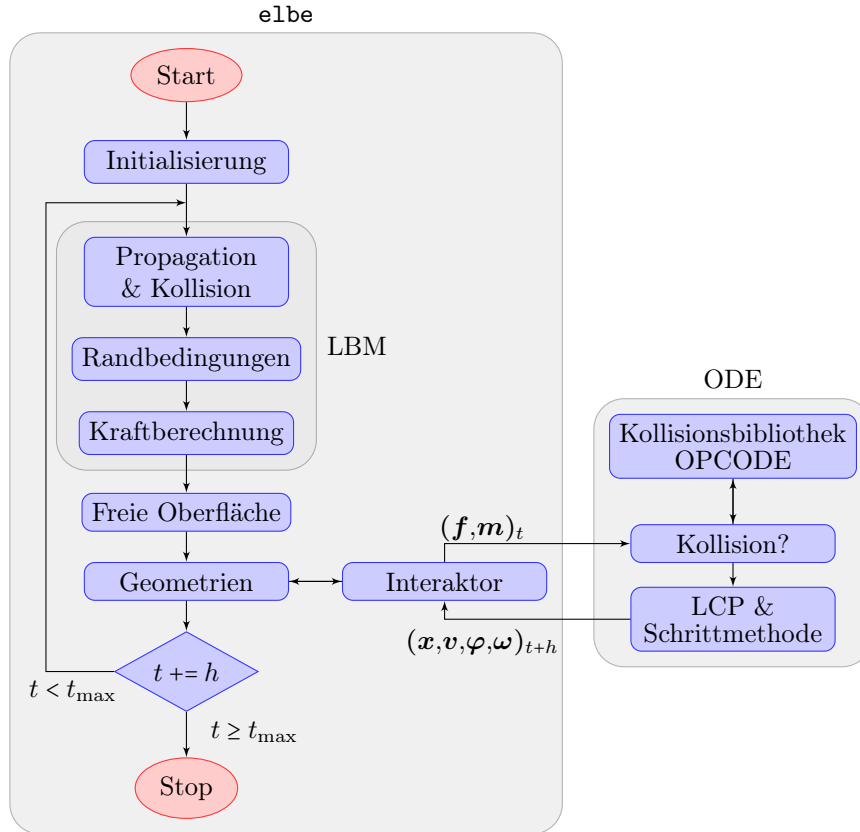


Abbildung 18: Schema der expliziten und bidirektionalen Fluid-Struktur-Simulation mit dem Strömungslöser `eibe` und dem Starrkörperlöser *Open Dynamics Engine* (ODE). Der Interaktor fungiert als Schnittstelle und übermittelt Fluidkräfte \mathbf{F} und Fluidmomente \mathbf{M} zum Zeitschritt t an den Starrkörperlöser. Dieser bestimmt auf Basis eventueller Körperkollisionen die resultierenden translatorischen und rotatorischen Positionen und Geschwindigkeiten der beteiligten Körper \mathbf{x} , \mathbf{v} , φ und ω zum Zeitschritt $t + \Delta t$.

Die Klasse `MCOpenDynamicsEngine` ist soweit gestaltet, dass sich der Benutzer keine Sorgen um die Initialisierung der richtigen ODE-Geometrie machen muss. Ein Aufruf der Funktion `addBody()` genügt, um in ODE einen Körper hinzuzufügen. Als Parameter werden die kinematischen Startbedingungen, Körperträgheit und Freiheitsgrade, sowie die `eibe`-Geometrie übergeben. Anschließend wird automatisch die entsprechende ODE-Geometrie aus den Informationen der `eibe`-Geometrie generiert. Selbst triangulierte Gittergeometrien werden in ODE selbstständig initialisiert, indem die Daten der Geometrieinstanz `G03DSurfaceMesh` entnommen werden, die zuvor aus dem `pip`-Dateiformat [54] gelesen wurden.

Zur Lösung des LCP wird die direkte Methode `WorldStep` verwendet. Als Hüllkörperhierarchie-

Methode wird `SAPSpace` gewählt. Wenn nicht anders erwähnt, wird das Reibungsmodell №1 mit $\mu = 0,5$ verwendet, das den Reibungskegel durch eine Reibungspyramide approximiert. Für die globalen Korrekturparameter gilt $ERP = 0,2$ und $CFM = 1 \cdot 10^{-10}$. Bei den Kollisionsgelenken wird $ERP_K = 0,1$ und $CMF_K = 1 \cdot 10^{-15}$ gewählt. Techniken zur Parallelisierung von ODE oder zur automatischen Deaktivierung passiver Körper werden nicht verwendet. Alle weiteren Parameter werden in ihrer Standardkonfiguration belassen.

In der Praxis konnte durch Skalierungstests festgestellt werden, dass die in der ODE verfügbaren Methoden hinreichend leistungsstark sind, um Simulationen mit bis zu 100 Schollen effizient zu bestimmen. Etwaige Optimierungen oder Erweiterungen des Kollisionsdetektions- oder -lösungsverfahrens waren daher nicht erforderlich.

4.4 Validierung

In diesem Abschnitt soll die erfolgreiche Implementierung der *Open Dynamics Engine* in `elbe` demonstriert und die Kernmethoden der ODE getestet werden. Hierzu sollen kleine Versuche geführt werden, deren Ergebnisse analytisch bekannt sind oder sich experimentell ermitteln lassen.

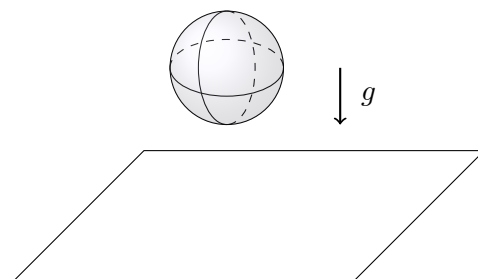
Simulationen der Starrkörperdynamik, die keine Fluidphase voraussetzen und somit *trocken* erfolgen, werden außerhalb eines minimalen Rechengebiets der Lattice-Boltzmann-Methode mit $10 \times 10 \times 10$ Gitterknoten geführt. Der Berechnungsaufwand der Fluidphase fällt somit vernachlässigbar klein aus und nimmt keinen Einfluss auf die Starrkörperdynamik.

4.4.1 Kollisionsbehandlung

Zur Validierung der *Open Dynamics Engine* soll in diesem Versuch auf Basis der zentralen Stoßmechanik das Verhalten der Kollisionserkennung und -behandlung in einer fortlaufenden Simulation untersucht werden. Hierfür wird ein Basketball der Masse 1 kg und dem Radius 0,5 m betrachtet, der sich $H = 1$ m lichte Höhe über dem starren Boden befindet und dem Schwerfeld $g = 9,81 \text{ m/s}^2$ ausgesetzt ist. Weitere Simulationsparameter finden sich in Abb. 19.

Parameter	Wert
Masse m	1 kg
Radius r	0,5 m
Initiale Schwerpunkthöhe z_0	1,5 m
Gavitation g	$9,81 \text{ m/s}^2$
Reibungskoeffizient μ	-
Restitutionskoeffizient k	$\in \{0,0; 0,5; 1,0\}$
Zeitschrittweite Δt	10^{-3} s

(a) Simulationsparameter



(b) Simulationsskizze

Abbildung 19: Details der Testumgebung zur Validierung der Kollisionserkennungs und -behandlung der *Open Dynamics Engine*.

Um jeweils ein vollplastisches, teilplastisches und vollelastisches Stoßverhalten zu erzielen, wer-

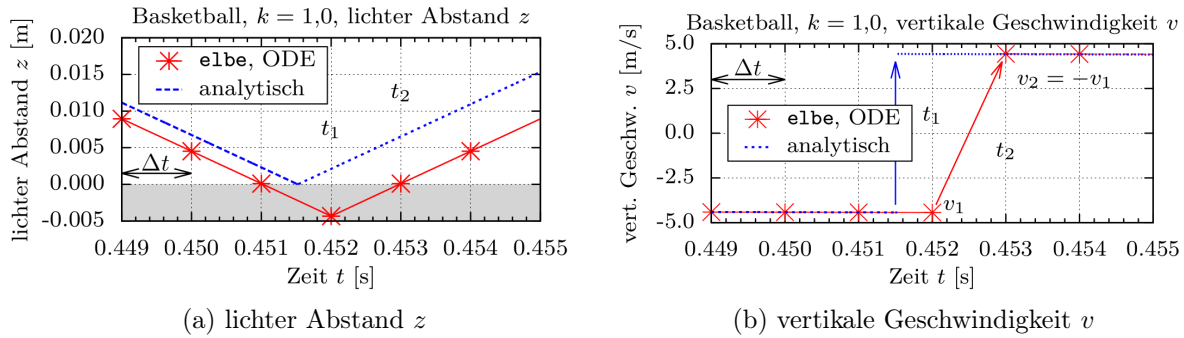


Abbildung 20: Kollisionsbehandlung, Basketball, vollelastischer Stoß ($k = 1,0$), erster Kollisionszeitraum.

den drei Simulationen mit unterschiedlichen Restitutionskoeffizienten $k \in \{0,0; 0,5; 1,0\}$ geführt. Für die Zeitschrittweite und Simulationsdauer gilt $\Delta t = 0,001$ s und $t_{\max} = 10$ s. Der Ball wird durch eine triangulierte Gittergeometrie aus 5120 gleichgroßen Dreiecken diskretisiert (im Engl. als *icosphere* bekannt). Ein triangulierter Würfel dient als starrer Untergrund.

Wie zu erwarten, setzt beim **vollelastischen Stoß** mit $k = 1,0$ ein parabelförmiges Oszillieren der Höhenlage ein. Die Periode $T = 2\sqrt{\frac{2H}{g}}$ wird lediglich um etwa 0,3 % überschätzt. Die Überlappungstiefe zwischen Ball und Boden bleibt unter $d \leq 0,005$ m.

Die Abb. 20 zeigt, um die Zeit der ersten Kollision herum, den lichten Abstand z zwischen Ball und Boden, sowie die vertikale Geschwindigkeit v des Balls. Zum Vergleich sind die analytischen Lösungen eingezeichnet, die aus dem Impuls- und Energieerhaltungssatz resultieren. In beiden Unterabbildungen 20a und 20b ist für ODE zu erkennen, dass die erste Geometrieüberlappung bei $t_1 = 0,452$ s stattfindet und im darauffolgenden Schritt zu $t_2 = 0,453$ s erkannt und behandelt wird. Diese verspätete Kollisionserkennung und -behandlung sorgt in dem betrachteten Szenario jeweils für eine Zufuhr zusätzlicher kinetischer Energie. Zum einen erfolgt die Umkehrung der Geschwindigkeit um einen Zeitschritt verspätet zu t_2 statt t_1 . Da diese Umkehrung $v_2 = -v_1$ numerisch exakt erfolgt, werden innerhalb dieses Zeitraums fälschlicherweise auch keine externen Kräfte berücksichtigt. Die Geschwindigkeitserhöhung beträgt somit $\Delta v_2 = g \Delta t$. Zum Anderen gebietet die verspätete Kollisionserkennung dem Ball eine leicht verlängerte Fallstrecke von $H + d$ und damit eine Erhöhung der Geschwindigkeit um $\Delta v_1 \leq g \Delta t$ zum Schritt t_1 , unmittelbar vor der Kollisionserkennung. Letztlich besitzt der Ball nach Abschluss der Kollisionsbehandlung bei t_2 eine um $g \Delta t \leq \Delta v \leq 2g \Delta t$ erhöhte Geschwindigkeit. Somit wird dem System während jeder Kollisionsauflösung eine kleine Menge kinetischer Energie zugeführt. In Abb. 21 lässt sich dies genauer erkennen. Sie stellt den Verlauf der normalisierten Gesamtenergie über die Simulationszeit dar. Überraschenderweise entspricht die zugeführte Energie genau der Menge, die während der Flugphase aufgrund der zeitlichen Diskretisierung dissipiert. Der Verlauf ähnelt somit einem Sägezahnprofil.

Die Simulation des **vollplastischen Stoßes** mit $k = 0,0$ verhält sich wie zu erwarten. Der Körper kommt nach der halben Periode, die, wie bereits ermittelt, mit etwa 0,3 % im Verzug liegt, schlagartig zur Ruhe. Aufgrund der Gravitationskraft bleibt in der weiteren Simulation stets

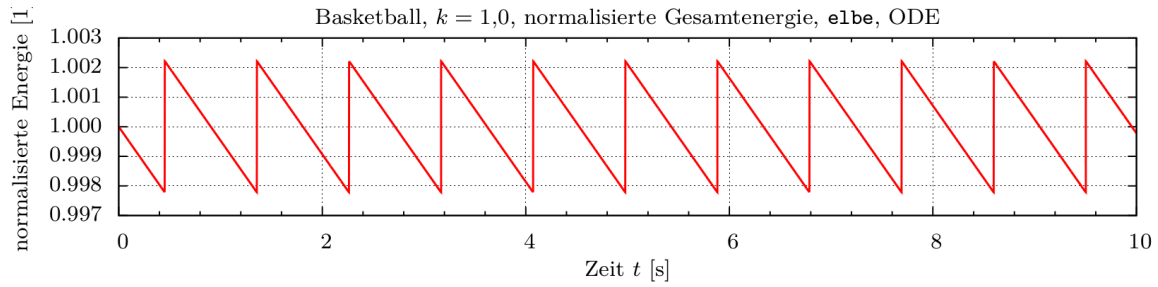


Abbildung 21: Kollisionsbehandlung, Basketball, vollplastischer Stoß ($k = 1,0$). Die normalisierte Gesamtenergie gleicht einem Sägezahnprofil.

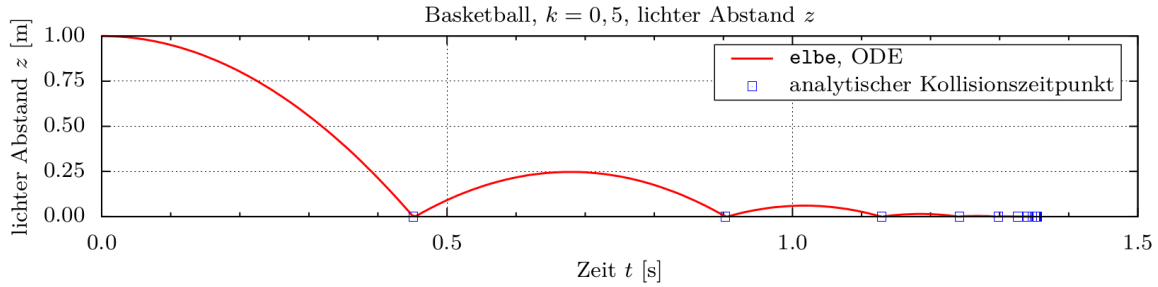


Abbildung 22: Kollisionsbehandlung, Basketball, teilplastischer Stoß ($k = 0,5$), lichter Abstand z .

eine Geometrieüberlappung bestehen. Die Eindringtiefe liegt zu Beginn mit ungefähr $0,00275$ m geringfügig höher als zuvor im vollplastischen Fall, reduziert sich nach kurzer Zeit jedoch auf $1 \cdot 10^{-3}$ m und behält diese Tiefe bei. Eine weitere Minimierung der Tiefe lässt sich durch das Herabsetzen des Parameters `ContactSurfaceLayer` erzielen. Er beschreibt die erlaubte Überlappungstiefe zweier Körper, die sich in Ruhelage befinden und dient normalerweise zur Prävention von *jittering*-Effekten. Es liegt somit in der Präferenz des Benutzer, welche Eindringtiefe in der Ruhelage akzeptabel ist und ob eventuelle *jittering*-Effekte auf das Möglichste vermieden werden sollen.

Der **teilplastische Stoße** mit der Stoßzahl $k = 0,5$ wird ebenfalls den Erwartungen gerecht. In Abb. 22 ist der lichte Abstand des Balls zum Boden über die Simulationszeit abgetragen. Die vertikale Geschwindigkeit halbiert sich unmittelbar nach der Kollisionsauflösung: $v_{\text{nach}} = k v_{\text{vor}}$. Dadurch reduziert sich die Gesamtenergie um $k^2 = 1/4$, was wiederum in einer Viertelung der Maximalhöhe resultiert. Zur Ruhe kommt der Ball schließlich bei $t = 1,35$ s mit einer Kollisionstiefe von $2,3 \cdot 10^{-7}$ m. Zum Vergleich sind in der Abbildung die analytischen Kollisionszeitpunkte eingetragen.

4.4.2 Reibungsmodelle

Die folgenden beiden Versuche dienen zur Untersuchung der in ODE integrierten Reibungsmodelle. In Kapitel 4.1.2 wurden diese bereits erläutert. Sie beschreiben das COULOMBSche Reibungsgesetz $|f_T| \leq \mu |f_N|$ approximativ und arbeiten zusammengefasst wie folgt:

- Modell №1: Approximation des Reibungskegels durch eine Reibungspyramide.
- Modell №2: μ dient als max. tang. Reibungskraft, unabhängig der Normalkraft.

Im Folgenden werden die Versuche von Seugling/Rölin [86] für beide Reibungsmodelle mit der in `elbe` implementierten aktuelleren ODE Version wiederholt.

Eindimensionaler Reibungsversuch In diesem Versuch soll untersucht werden, wie sich die in ODE auftretenden Reibungskräfte zum COULOMBSchen Reibungsgesetz verhalten. Dazu soll das Kräfteverhältnis

$$\frac{|f_T|}{|f_N|} \quad (28)$$

an einer sich kontinuierlich neigenden Ebene betrachtet werden, auf der ein Quader aufliegt. Für $\frac{|f_T|}{|f_N|} < \mu$ sorgt die Selbsthemmung bekanntlich für eine Haftreibung zwischen Quader und Ebene. Der Fall $\frac{|f_T|}{|f_N|} = \mu$ beschreibt den Übergang von Haft- zu Gleitreibung. Er kennzeichnet sich durch $\tan \theta = \mu$, wobei θ die Neigung der Ebene aus dem Lot beschreibt, bei der der Quader die Haftung verliert und in Relativbewegung gerät. Um Koordinatentransformationen zu vermeiden und einen geringeren Berechnungsaufwand zu erzielen, wird statt einer Neigung der Ebene, eine Variierung der Richtung des Gewichtskraftvektors vorgenommen

$$\mathbf{f}_g = -m g \begin{pmatrix} 0 & \sin \theta & \cos \theta \end{pmatrix}^T . \quad (29)$$

Die tangentielle Reibungs- und Normalkraft, f_T und f_N , lassen sich anschließend aus den globalen Kollisionskräften extrahieren

$$\mathbf{f}_K = \begin{pmatrix} 0 & f_T & f_N \end{pmatrix}^T . \quad (30)$$

Die Masse des Würfels sei mit 1 kg angenommen. Für die Gravitationskonstante gelte $g = 10 \text{ m/s}^2$ und der Haftreibungskoeffizient wird für beide Reibungsmodelle auf $\mu = 0,5$ gesetzt (im Weiteren soll vernachlässigt werden, dass dieser Parameter für das Modell № 2 eine völlig andere Bedeutung hat, siehe Kapitel 4.1.2). Die Simulation erfolgt mit einer Zeitschrittweite von $\Delta t = 0,001 \text{ s}$. Für die Stoßzahl gilt $k = 0,5$. Die Auslenkung der Gewichtskraft \mathbf{f}_g soll zunächst $\theta = 0^\circ$ betragen und ab Schritt 100 mit jedem weiteren Schritt um $0,001^\circ$ erhöht werden, bis $\theta = 90^\circ$ erreicht ist oder der Quader den Kontakt zur Ebene komplett verliert. Quader und Ebene werden jeweils durch eine triangulierte Gittergeometrie approximiert.

In Abb. 23 ist das Kräfteverhältnis $\frac{|f_T|}{|f_N|}$ für beide Reibungsmodelle über den Tangens der Gewichtskraftauslenkung θ als COULOMBSche Reibungskurve abgetragen. Der Übergang von Haft- zu Gleitreibung ist somit bei $\tan \theta = \mu = 0,5$ zu erwarten. Um den Zeitpunkt der ersten Relativbewegung herauszufinden, ist in Abb. 24 die Relativgeschwindigkeit des Quaders zur Ebene logarithmisch über die Gewichtskraftauslenkung $\tan \theta$ abgetragen. In beiden Abbildungen lässt sich erkennen, dass sich die Ergebnisse der Reibungsmodelle stark unterscheiden.

Das Modell №1 zeigt ein sehr gutes Verhalten und kommt in diesem Szenario dem COULOMBSchen Reibungsgesetz sehr nahe. Mit nur einigen kleinen Ausnahmen steigt das Kräfte-

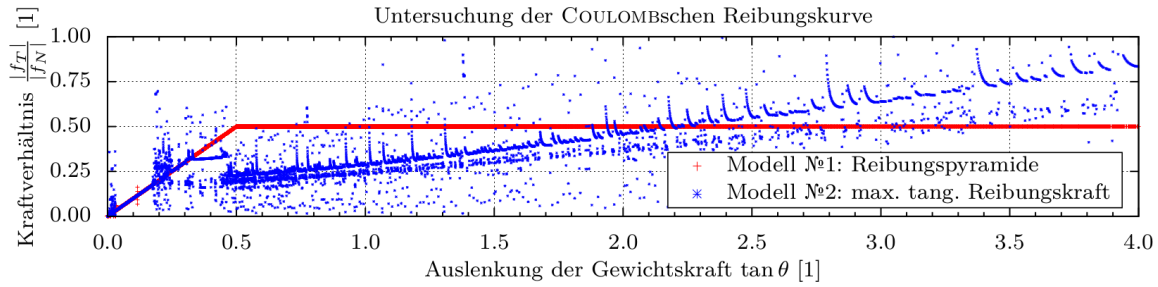


Abbildung 23: Eindimensionaler Reibungsversuch, Untersuchung der COULOMBSchen Reibungskurve.

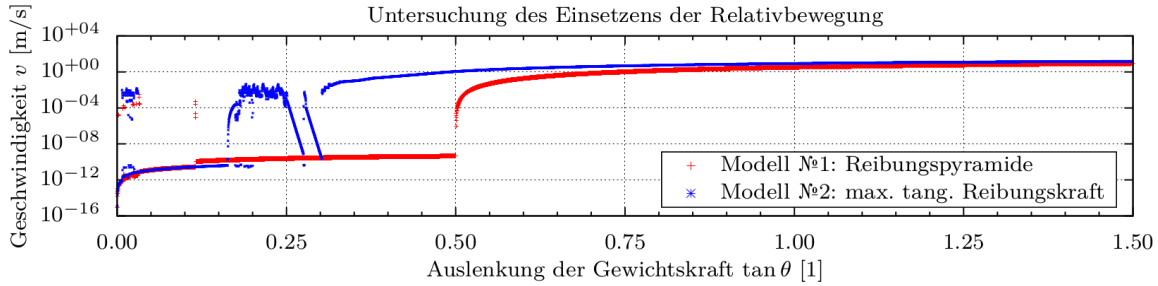


Abbildung 24: Eindimensionaler Reibungsversuch, Untersuchung der Relativgeschwindigkeit.

verhältnis $\frac{|f_T|}{|f_N|}$ proportional zum Tangens der Neigung an. Erst ab $\tan \theta = \mu = 0,5$ nimmt das Kraftverhältnis einen konstanten Verlauf an und der Quader beginnt zu gleiten, wie auch in Abb. 24 zu erkennen ist. Lediglich einige kleine Ausreißer lassen den Quader kurzzeitig zu früh gleiten. Insgesamt sieht man dem Modell die Approximation durch eine Reibungspyramide nicht an. Dies liegt daran, dass die Auslenkung parallel zu einer Kante der Pyramide erfolgt und damit den selben radialen Abstand zur Normalen aufweist wie im Falle eines Reibungskegels.

Bei Modell №2 zeigt sich ein physikalisch mehrheitlich inkorrektes und schwer nachvollziehbares Verhalten. Das Kräfteverhältnis $\frac{|f_T|}{|f_N|}$ weist einige größere Ausschläge von bis zu 6 auf, die in Abb. 23 außerhalb des Zeichenbereichs liegen. Lediglich im Auslenkungsbereich um $\tan \theta \approx 0,1$ herum zeigt sich ein proportionales Verhalten wie bei Modell №1. Ähnliches lässt sich aus Abb. 24 ablesen. Erstes zeitweises Rutschen setzt bereits kurz nach Beginn der Auslenkung bei etwa $\tan \theta = 0,008$ ein und dauerhaftes Rutschen ab $\tan \theta = 0,3$. Im Zuge der starken Approximation wird das COULOMBSche Reibungsgesetz $|f_T| \leq \mu |f_N|$ nicht einhalten.

Zweidimensionaler Reibungsversuch Aufgrund der starken Vereinfachung ist bei beiden Reibungsmodellen ein anisotropes Verhalten zu erwarten. Dies soll im Folgenden zweidimensionalen Reibungsversuch untersucht werden.

Der vorherige Versuchsaufbau wird soweit erweitert, dass es möglich ist, die Ebene in alle Richtungen neigen zu können. Anschließend soll untersucht werden, unter welchem Neigungswinkel θ und welcher Neigungsrichtung φ der Quader zu rutschen beginnt.

Statt die Ebene zu neigen, wird abermals die Richtung der Gravitationsbeschleunigung \mathbf{f}'_g va-

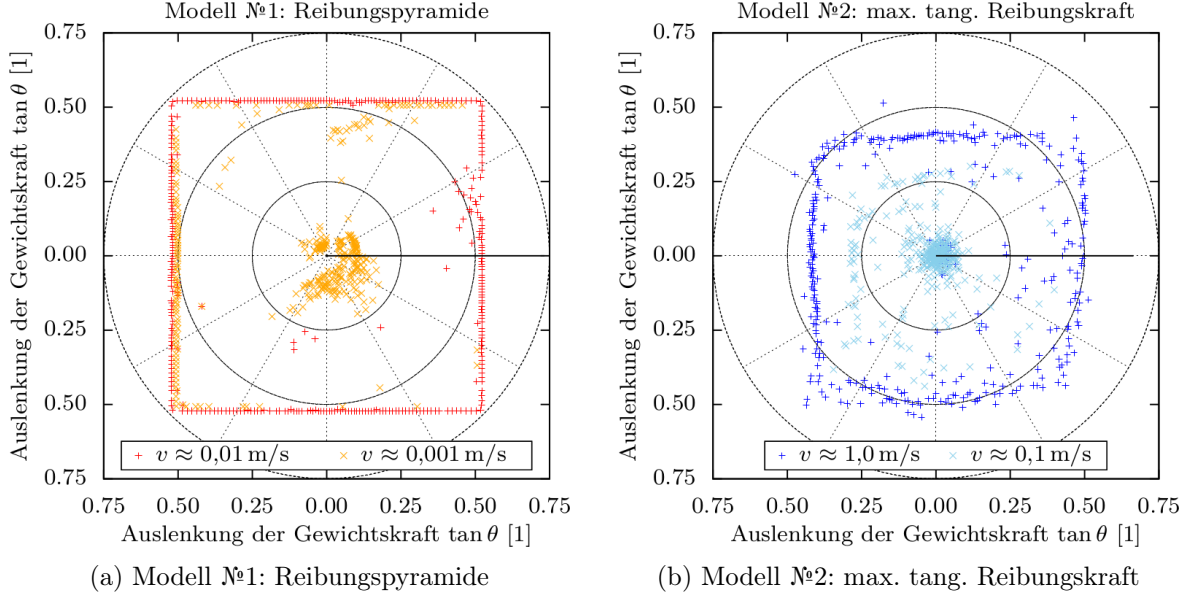


Abbildung 25: Zweidimensionaler Reibungsversuch, Untersuchung auf Anisotropie.

riert. Hierzu dient die Rotationsformel von Rodrigues

$$\mathbf{f}'_g = \mathbf{f}_g \cos \theta + (\mathbf{n} \times \mathbf{f}_g) \sin \theta + \mathbf{n}(\mathbf{n} \cdot \mathbf{f}_g)(1 - \cos \theta), \quad (31)$$

mit \mathbf{f}_g als lotrechte Gravitationskraft und \mathbf{n} als Drehachse der Länge eins

$$\mathbf{f}_g = -m g \mathbf{e}_z \quad \text{und} \quad \mathbf{n} = \begin{pmatrix} -\cos \varphi & \sin \varphi & 0 \end{pmatrix}^T. \quad (32)$$

Es werden 360 Simulationen geführt, in denen die Neigungsrichtung φ von 0 bis 2π variiert wird. Die Simulationsparameter gleichen denen des vorherigen Versuchs. Sobald eine bestimmte Relativgeschwindigkeit überschritten wird, erfolgt eine vorzeitige Beendigung der Simulation.

Abb. 25 stellt die Ergebnisse beider Reibungsmodelle in Polardiagrammen dar. Über den Radius ist die Auslenkung $\tan \theta$ der Gewichtskraft \mathbf{f}'_g abgetragen und der Polarwinkel kennzeichnet die Neigungsrichtung φ der Ebene. Wie der vorherige Versuch bereits in Abb. 24 zeigte, ist es schwierig zu beurteilen, ab welchem Auslenkungswinkel die Relativbewegung des Quaders einsetzt. Im Folgenden soll daher nicht das langsame zeitweise Rutschen untersucht werden, sondern nur ab welchem Winkel ein *dauerhaftes* Gleiten vorliegt. Um dies zu erzielen, wird aus Abb. 24 abgelesen, ab welcher Geschwindigkeit ein ununterbrochenes Gleiten vorliegt. Für Modell №1 liegt diese Geschwindigkeit bei etwa 0,01 m/s und für Modell №2 bei etwa 1 m/s. Um das Problem zu verdeutlichen, soll jeweils zusätzlich gegen eine weitere Geschwindigkeit getestet werden, die eine Größenordnung kleiner ausfällt. Diese Untersuchungsmethode nimmt zwar viele Vereinfachungen in Kauf, zur Beurteilung der Isotropie ist sie jedoch ausreichend gut.

Wie zu erwarten, lässt sich in Abb. 25a erkennen, dass das Reibungsmodell №1 anisotropie in Form eines Rechtecks aufweist, was zwangsweise aus der Reibungspyramide resultiert. Neigt sich

die Ebene lediglich in eine Hauptachsenrichtung, so findet der Übergang von Haft- zu Gleitreibung zum erwarteten Auslenkungswinkel $\tan \theta = \mu = 0,5$ statt und kommt dem COULOMBSchen Modell sehr nahe. Liegt jedoch eine Neigung entlang einer Achsenhalbierenden vor, dann setzt das Gleiten erst bei einer größeren Neigung ein. Dieses Verhalten gleicht dem COULOMBSchen Modell, wenn dessen Haftreibungskoeffizient μ um den Faktor $\sqrt{2}$ größer angesetzt wird. Demnach können tangentielle Reibungskräfte bis zu einem Faktor von $\sqrt{2}$ bzw. etwa 41 % überschätzt werden.⁸

Das Reibungsmodell №2 weist in Abb. 25b ebenfalls ein anisotropes Verhalten auf, das im Vergleich zu Modell №1 etwas geringer ausfällt. Allerdings ist zu beachten, dass Abb. 25b nur bedingt aussagekräftig ist. Aufgrund der großer Unstetigkeiten und dem frühzeitigen und zeitweisen Rutschen des Quader (siehe Abb. 24), sowie der relativ hohen Geschwindigkeit, die als Grundlage zur Abschätzung des Einsetzens des ununterbrochenen Gleiten dienen muss, soll dieses Modell nicht weiter untersucht werden. In den weiteren Simulationen ist das Modell №1 vorzuziehen.

Die präsentierten Ergebnisse fallen ähnlich zu denen von Seugling/Rölin [86] von 2006 aus, die sich auf das Modell №1 beschränkten. Daraus lässt sich schließen, dass ODE seitdem keine signifikanten Erweiterungen oder Verbesserungen am Reibungsmodell erfuhr.

4.4.3 Kollisionsaufwand

In diesem Versuch soll der Berechnungsaufwand der Kollisionsmethode untersucht werden. Betrachtet werden zwei Körper mit der selben triangulierten Gittergeometrie. Ein Körper sei starr und der andere bewege sich aufgrund von Gravitation auf den ersten zu. Um die Körper nach der ersten Kollision nahe beieinander zu behalten, wird nahezu voll-plastisches Verhalten mit der Stoßzahl $k = 0,01$ eingestellt. Als Zeitschrittweite wird $\Delta t = 1 \cdot 10^{-5}$ s gewählt. Die Gitter beider Körper bestehen zunächst aus jeweils 10960 Dreiecken und werden in vier weiteren Simulationen mit den Faktoren $2^2, 3^2, 4^2, 5^2$ quadratisch verfeinert. Während der Simulation wird jeweils die Berechnungsdauer von drei Berechnungsschritten gemessen: die der Kollisionserkennung durch OPCODE, die des LCP-Lösungs- und Schrittverfahrens von ODE und die restliche Zeit, die von `e1be` beansprucht wird, um den Zeitschritt abzuschließen und den nächsten bis zur Kollisionserkennung einzuleiten.

Abb. 26 zeigt die Berechnungsdauer der Kollisionserkennung in Millisekunden über die Simulationszeit. Die Fernphase lässt sich daran erkennen, dass nahezu kein Berechnungsaufwand anfällt. Die Hüllkörper beider Körper überlagern sich in dem Bereich $0,18\text{s} < t < 0,42\text{s}$, der zugleich die Nahphase beschreibt und in dem die eigentliche Kollisionserkennung durch OPCODE erfolgt. Dennoch findet erst im Bereich $0,22\text{s} < t < 0,4\text{s}$ eine Steigerung des Berechnungsaufwands statt. Genau zu diesen lokalen Maxima finden auch die Geometrikollisionen statt, die in gelb hinterlegt sind.

⁸Mit dieser Kenntnis biete sich an, den Simulationsparameter μ soweit geringer wählen, dass ein über alle Richtungen ausgewogener Fehler erzielt wird (vergl. *Quadratur des Kreises*).

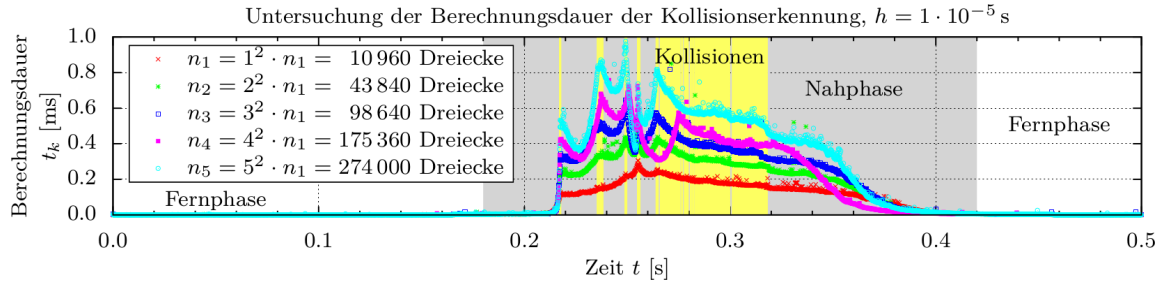


Abbildung 26: Untersuchung der Berechnungsdauer der Kollisionserkennung.

Die Ergebnisse der fünf Simulationen sind untereinander nahezu äquidistant. Aufgrund des quadratischen Verfeinerungsfaktors bedeutet dies, dass der Berechnungsaufwand ähnlich der Wurzelfunktion der Dreiecke n wächst: $\mathcal{O}(\sqrt{n})$. Laut der Dokumentation von OPCODE [98], weist die Kollisionsbibliothek starke Optimierungen auf und nutzt zur beschleunigten Kollisionserkennung Informationen des letzten Zeitschritts aus. Eine Regressionsanalyse an der Stelle $t = 0,22 \text{ s}$ führt mit $t_k(n) = a n^b$ auf $a = 0,00207$ und $b = 0,4373$, wobei t_k den Rechenaufwand in Millisekunden beschreibt.

Da eine Gitterverfeinerung keinen Einfluss auf die Dimension des LCP hat, fällt der Aufwand zur Lösung und die des Zeitschrittverfahrens für alle Simulationen nahezu konstant aus mit $t \approx 0,01 \text{ ms}$.

Der restliche Berechnungsaufwand, der den Zeitschritt abschließt und den nächsten bis zur Kollisionserkennung einleitet, fällt im Vergleich zu den vorherigen Ergebnissen wesentlich größer aus. Die fünf simulierten Fälle benötigen 2; 5,5; 9,5; 17,5 und 25 ms je Zeitschritt. Die Berechnungszeit steigt somit fast linear mit der Anzahl der beteiligten Dreiecke. Anzumerken ist, dass zum Zeitpunkt des Versuches noch keine Optimierungen zur Simulation vieler Körper an `elbe` vorgenommen wurden.

4.4.4 Zusammenfassung

Die in Kapitel 4.4.1 untersuchten zentralen Stöße eines Basketballs zeigen sehr gute Ergebnisse im voll- und teilplastischen Verhalten. Der vollelastische Stoßversuch resultiert ebenfalls in einem sehr guten Ergebnis. Die Auswertung zeigte außerdem, dass dem System kinetische Energie zugeführt wird, wenn zum Kollisionszeitpunkt die eingprägten Kräfte und die Körpergeschwindigkeit gleichgerichtet sind. Sind die eingprägten Kräfte der Körpergeschwindigkeit entgegengerichtet, so wird mit einer leichten sprunghaften Reduktion der kinetischen Energie zu rechnen sein. Beides resultiert aus einer verspäteten Kollisionserkennung und -behandlung.

In Kapitel 4.4.2 wurden die in die *Open Dynamics Engine* implementierten Reibungsmodelle getestet, die einer starken Vereinfachung des COULOMBSchen Reibungsgesetzes entstammen. Es zeigte sich, dass das Modell №1, das auf einer Reibungspyramide basiert, dem anderen Modell vorzuziehen ist und nur selten ein vorzeitiges Gleiten erlaubt. Je nach Reibungsrichtung kann die tangentielle Reibungskraft um den Faktor $\sqrt{2}$ überschätzt werden. Entsprechend verzögert sich

das Einsetzen des Rutschens. Das Reibungsmodell №2, das lediglich eine maximale tangentielle Reibungskraft als Parameter bietet, weist viele Unstetigkeiten auf und erlaubt ein frühzeitiges und zeitweises Gleiten (siehe Abb. 24). Bei beiden Modellen ist mit Anisotropie zu rechnen.

Der Kapitel 4.4.3 zeigte, dass in der Nahphase zweier kollidierender triangulierter Gittergeometrien der Berechnungsaufwand der Kollisionserkennung bei einer Gitterverfeinerung ähnlich der Wurzelfunktion der Dreieckanzahl steigt. In der Fernphase fällt der Berechnungsaufwand vernachlässigbar gering aus. Da der restliche Berechnungsaufwand eines Zeitschritts in etwa linear mit der Dreiecksanzahl steigt, relativiert sich der Aufwand für die Kollisionserkennung.

Insgesamt lässt sich sagen, dass die *Open Dynamics Engine* auf einem effizienten und ausreichend genauem Kollisionsdetektions- und -lösungsverfahren, sowie einem hinnehmbaren Reibungsmodell basiert. Die *Open Dynamics Engine* wird den Anforderungen dieses Forschungsprojekts an eine Physik-Bibliothek zur Simulation der Kollision vieler Körper gerecht.

5 Bidirektionale Fluid-Struktur-Interaktion

Zur Simulation der Interaktion zwischen Fluid und Starrkörperstrukturen wurde eine Schnittstelle zur Kopplung des Starrkörperlösers *Open Dynamics Engine* an den Fluidlöser **elbe** geschaffen. Für die Schnittstelle, dem sogenannten Interaktor, wurde ein expliziter, partitioniert monolithischer und bidirektionaler Ansatz gewählt, der bereits in [60, 36, 42, 37] erfolgreich zur Kopplung einer Physik-Bibliothek an einen Gitter-Boltzmann Strömungslöser angewendet wurde. Strömungs- und Starrkörperlöser sind strikt von einander getrennt und kommunizieren lediglich über den Interaktor miteinander. Dennoch übernimmt der Strömungslöser mittels des Interaktors die steuernde Funktion über den Starrkörperlöser.

Während eines Zeitschritts ermittelt der Strömungslöser die auf die Starrkörpergeometrien wirkenden integralen hydrodynamischen Lasten, die daraufhin mittels des Interaktors an den Starrkörperlöser übertragen werden. Anschließend bestimmt dieser auf Basis der Massenträgheiten und eventueller Körperkollisionen die aus den Fluid- und weiteren externen Lasten, wie bspw. der Gravitation, resultierenden Verschiebungen und Geschwindigkeiten der Starrkörper und sendet diese zurück an den Strömungslöser. Dieser aktualisiert daraufhin die Körperkinematik im Fluidfeld und leitet den nächsten Zeitschritt ein.

Die im Projektantrag erwähnte Möglichkeit der Nutzung eines zeitlichen Subcyclings des Starrkörperlösers, um eine eventuell erhöhte Dynamik der Strömungshindernisse ausreichend genau modellieren zu können, stellte sich in kleinen Tests als eher nachteilig heraus und war bisher auch nicht erforderlich. Der Ansatz wurde daher nicht weiter verfolgt.

Im Folgenden werden die Verfahren zur Bestimmung der hydrodynamischen Lasten erläutert.

5.1 Kraftberechnungsverfahren

Zur Bestimmung der auf ein Strömungshindernis wirkenden integralen Fluidlast sind mehrere Schritte erforderlich. Zunächst müssen die Teillasten an allen Gitterknoten berechnet werden, die sich in unmittelbarer Nachbarschaft zur Hindernisgeometrie befinden. Dabei ist es ausreichend lediglich die Fluidgitterknoten zu berücksichtigen, die in der konservativ auf die Gitterknoten projizierten AABB der betrachteten Geometrie liegen. Hierzu bieten sich zwei Methoden an: die Impulsaustauschmethode und die Integration der Drücke bzw. Spannungen. Letztere kann wahlweise über alle beteiligten Gitterknoten oder über die Dreiecke der triangulierten Gittergeometrie erfolgen und bietet entsprechende Vor- und Nachteile. Für einen detaillierten Vergleich beider Kraftberechnungsmethoden sei auf Mei *et al.* [69] verwiesen. Nach Berechnung der Teillasten ist es erforderlich diese effizient aufsummiert. Die Verfahren zur Kraftberechnung und anschließenden Addition werden im Folgenden präsentiert.

Unabhängig von der Kraftberechnungsmethode bestimmt sich das auf das Hindernis wirkende Teilmoment für jeden Gitterknoten mittels $\mathbf{M} = \mathbf{h} \times \mathbf{F}$, wobei \mathbf{h} vereinfacht den Hebelarm vom Schwerpunkt des Hindernisses zum jeweiligen Gitterknoten und \mathbf{F} die Teilfluidlast beschreiben.

5.1.1 Impulsaustauschmethode über die Gitterknoten

Bei der Impulsaustauschmethode (engl. *momentum exchange method*, MEM) berechnet sich die Fluidlast an einem Fluidgitterknoten aus der Bilanzierung der Teilchenimpulse vor und nach dem Auftreffen auf die Hindernisberandung Γ

$$\mathbf{F} = \sum_{i \in \Gamma} \mathbf{F}_i = - \sum_{i \in \Gamma} \frac{V}{\Delta t} (\mathbf{e}_i f_i(t, \mathbf{x} + \Delta t \mathbf{e}_i) - \mathbf{e}_i f_I(t + \Delta t, \mathbf{x})) \quad (33)$$

$$= - \sum_{i \in \Gamma} \frac{V}{\Delta t} \mathbf{e}_i (f_i(t, \mathbf{x} + \Delta t \mathbf{e}_i) + f_I(t + \Delta t, \mathbf{x})), \quad (34)$$

wobei $V = \Delta x^D$ das Zellvolumen in D Raumdimensionen beschreibt und i die Richtungen, welche die Berandung Γ schneiden, vgl. Nguyen und Ladd [79] sowie Bouizidi *et al.* [18]. Anzumerken ist, dass $f_i(t, \mathbf{x} + \Delta t \mathbf{e}_i)$ die propagierte und von der Randbedingung unbehandelte Verteilungsfunktion kennzeichnet und $f_I(t + \Delta t, \mathbf{x})$ die von der Randbedingung ermittelte und zurückgeführte Verteilungsfunktion darstellt.

Die Methode ist impulserhaltend und hinreichend genau, solange die Berandung Γ des Hindernisses mindestens einige Gitterknoten umfasst, wie Geller [36] feststellte. Genau genommen kennzeichnet Gleichung 33 die Fluidlast zum Kollisionszeitpunkt $t + \frac{1}{2}\Delta t$. Für genügend kleine Δt kann dies vernachlässigt werden.

Für bewegte Hindernisse ist ein zusätzlicher Korrekturterm erforderlich, der auch als Netto-Massenkorrektur (engl. *net mass correction*, MEMnmc) bezeichnet wird. Eine Möglichkeit ist die von Chen *et al.* [21] präsentierte Methode

$$\Delta \mathbf{F}_i = -6 w_i \rho \frac{\mathbf{e}_i \cdot \mathbf{u}_B}{c^2} \mathbf{e}_i, \quad (35)$$

die als sehr stabil gilt. Dabei kennzeichnet \mathbf{u}_B die Geschwindigkeit der Hindernisberandung an der von \mathbf{e}_i geschnittenen Stelle. Diese und weitere Korrekturmethode finden sich in Caiazzo und Jung [19], Nguyen und Ladd [79] sowie Geller [36].

5.1.2 Spannungsintegrationsmethode über die Gitterknoten

Die Spannungsintegrationsmethode (engl. *stress integration method*, SIM) verwendet den lokalen Spannungstensor $\boldsymbol{\sigma}$ zur Bestimmung der auf ein Hindernis wirkenden Teilfluidlasten. Im Gegensatz zu konventionellen CFD-Lösern lässt sich $\boldsymbol{\sigma}$ in der LBM lokal ermitteln. Die Spannung $\sigma_{\alpha\beta}$ kann aus den sogenannten Nichtgleichgewichtsverteilungen $f_i^{\text{neq}} = f_i - f_i^{\text{eq}}$ am Gitterknoten berechnet werden

$$\sigma_{\alpha\beta} = \left(1 - \frac{\Delta t}{2\tau}\right) \sum_{i=2}^n f_i^{\text{neq}} \left(e_{i\alpha} e_{i\beta} - \frac{1}{D} e_i^2 \delta_{\alpha\beta} \right) \quad \alpha, \beta \in \{1, 2, 3\}, \quad (36)$$

wobei δ das Kronecker-Delta, D die Raumdimension und n die Anzahl diskreter Richtungen ist sowie die Richtung $\mathbf{e}_1 = \mathbf{0}$ keine Berücksichtigung findet, vgl. Mei *et al.* [69]. Alternativ lässt

sich die Spannung $\sigma_{\alpha\beta}$ auch direkt aus den Momenten des *MRT*-Kollisionsoperator ermitteln. Anschließend lässt sich die auf die Hindernisberandung Γ wirkende Fluidkraft über

$$\mathbf{F} = \int_{\Gamma} \boldsymbol{\sigma} \cdot \mathbf{n} d\Gamma \quad (37)$$

bestimmen, wobei \mathbf{n} die Normale der Berandung beschreibt. Es sei erwähnt, dass Gleichung 36 genau genommen den Spannungstensor am Fluidknoten beschreibt. Zur exakten Bestimmung der Fluidlast ist jedoch der Spannungstensor auf der Hindernisberandung Γ erforderlich. Dieser ließe sich beispielsweise durch aufwendige Extrapolation bestimmen. Statt dessen wird die Teilfluidkraft über die Fläche integriert, die Fluid- und markierte Hindernisknoten trennt. Auf Extrapolation des Spannungstensors wird verzichtet und eine Bestimmung der Normale \mathbf{n} ist nicht mehr erforderlich

$$\mathbf{F} = \sum_{i \in \Omega} \boldsymbol{\sigma} \cdot \mathbf{e}_i, \quad (38)$$

wobei Ω die Indexmenge der Hauptrichtungen darstellt.

Näheres zur SIM findet sich in Yu *et al.* [111] und Geller [36]. Beide verdeutlichen außerdem, dass bei der Kraftberechnung von Hindernissen, die nur wenige Gitterknoten umfassen, SIM gegenüber MEM vorzuziehen ist.

5.1.3 Spannungsintegrationsmethode über die Geometrie

Je nach Anwendungssituation kann es ratsamer sein ein anderes Verfahren zur Spannungsintegration zu wählen. Beispielsweise ist die Spannungsintegration über die Gitterknoten relativ ungenau für Fälle, in denen sich Geometrien in Überlappungszonen verschiedener Gitterauflösungen oder sehr nah zueinander befinden.

Eine genaueres Ergebnis bietet die Spannungsintegration über die Dreiecke der Geometrien. Hierbei wird für jedes Dreieck ein Fluidknoten als Quelle zur Bestimmung der Spannung nach Gleichung 36 definiert. Dieser Quellknoten kann Beispielsweise der zum Dreiecksmittelpunkt am Nächsten gelegenen valide Fluidknoten sein. Anschließend wird die Spannung mit Hilfe der Normalen \mathbf{n} über die Dreiecksfläche A_{Δ} integriert

$$\mathbf{F} = A_{\Delta} \boldsymbol{\sigma} \cdot \mathbf{n}. \quad (39)$$

In einigen Fällen ist es von Vorteil Dreiecke und deren Quellknoten auszusparen, deren Anzahl an benachbarten validen Fluidknoten ein Minimum von zum Beispiel zwei Knoten nicht überschreiten, da diese häufig zu starken Ausreißern führen.

Auch bei dieser Spannungsintegrationsmethode wird auf eine im Allgemeinen genauere Extrapolation des Spannungstensors auf die Integrationsfläche A_{Δ} verzichtet. Da sich die auf ein Dreieck wirkende Fluidlast lediglich auf Basis eines Gitterknotens definiert, ist zu beachten, dass die Genauigkeit dieser Methode stark vom Auflösungsverhältnis zwischen dem Gitter und der Dreiecke

abhängt. Daher sollten die Dreiecke der Geometrie möglichst eine Kantenlänge besitzen, die in etwa der Gitterweite Δx entspricht.

6 Numerische Methoden zur Effizienzsteigerung

Während des Projektes wurde zahlreiche numerische Methoden entwickelt und implementiert, die einen bedeutenden Beitrag zur Effizienz der Simulation mit `elbe` beitrugen. Die bedeutendsten Methoden werden im Folgenden erläutert.

6.1 Indirekte Adressierung des freie-Oberflächen-Modells

Vor Projektbeginn arbeiteten die in dem freie-Oberflächen-Modell zur Anwendung kommenden Methoden aus Kapitel 2.5 aus Gründen der Vereinfachung vorsorglich auf allen Gitterknoten. Für eine Vielzahl an Gitterknoten war diese Vorgehensweise überflüssig und bot daher einsparbare Rechenkapazität. Um die Methoden nur auf den relevanten Gitterknoten bzw. Interface-Knoten wirken zu lassen, werden die betreffenden Gitterknoten in einem vorherigen Schritt herausgefiltert und den entsprechenden Methoden direkt übergeben. Ein solches Verfahren wird im Allgemeinen *Indirekte Adressierung* genannt. Die Indexierung der Interface-Knoten erfolgt parallelisiert zu jedem Zeitschritt und ist nur von geringem Aufwand. Je nach Systemkonfiguration ließ sich mit dieser Technik die Berechnungsdauer des freie-Oberflächen-Modells um $\mathcal{O}(10)$ verkürzen.

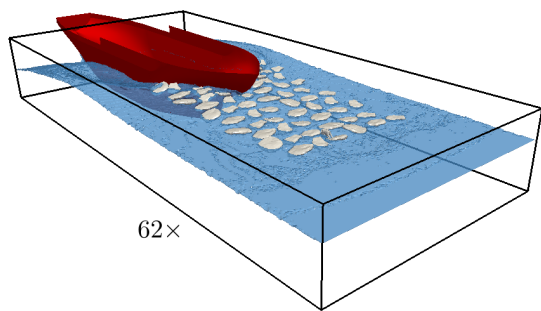
Zur Indexierung bzw. Filterung werden Methoden der CUDA-UnBound-Bibliothek (CUB) [70] verwendet. CUB ist eine low-level header-only Bibliothek, die speziell für CUDA von Nvidia Research entwickelt wurde und hoch parallelisierte Algorithmen zur Verfügung stellt.

6.2 Hüllkörper-basierte Hafttrandbedingung und Kraftberechnung

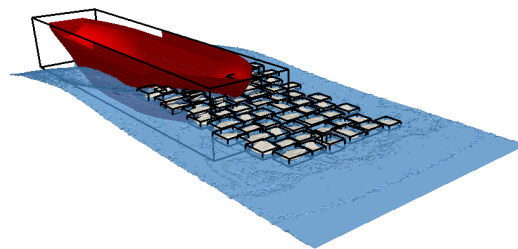
Ähnlich wie beim freie-Oberflächen-Modell, wurden Randbedingungen und Kraftberechnungsmethoden zuvor – der Einfachheit halber präventiv – auf allen Gitterknoten angewandt (auch wenn hierfür nur ein Bruchteil der Gitterknoten notwendig war). Die Methoden besaß damit eine lineare Komplexität in Abhängigkeit der Anzahl der Körper und der Anzahl der Gitterknoten des gesamten Rechengebietes, was sich unmittelbar in einer geringen Effizienz widerspiegelte. Da allerdings nur die erste geschlossene äußere Gitterknoten-hülle der Strömungshindernisse relevant ist, ließ sich auch dieses Verfahren beschleunigen. Eine Methode zum Filtern der tatsächlich beteiligten Gitterknoten war aus Daten-strukturellen und algorithmischen Gründen bisher nicht geglückt. Stattdessen konnten innovative und effiziente Hüllkörper-basierte Verfahren entwickelt werden, die es erlaubten die Menge der beteiligten Gitterknoten einer Methode geometrisch schärfer einzugrenzen.

Um die für die Methoden relevanten Gitterknoten schärfer einzugrenzen, wurden achsparallele Hüllkörper eingeführt und somit wird die Anzahl der vermeintlich beteiligten Gitterknoten erheblich reduziert, vgl. Abb. 27.

Als Hüllkörper werden AABB verwendet, da sich diese sehr einfach auf Gitterkoordinaten umrechnen lassen. Für bewegliche Strömungshindernisse müssen diese AABB zu jedem Zeitschritt neu berechnet werden. Im Falle primitiver Geometrien lassen sich die Eckpunkte der AABB mit



(a) Vorher: Methoden verrichten ihre Arbeit für jede Körpergeometrie auf jedem Gitterknoten des gesamten Rechengebiets, unabhängig des Bedarfs.



(b) Nachher: Methoden verrichten ihre Arbeit nur in den Hüllkörpern der Geometrien.

Abbildung 27: Beschleunigung der Hafttrandbedingung und Kraftberechnung durch Hüllkörper am Beispiel von $n = 62$ beweglichen Strömungshindernissen. Resultat: $\mathcal{O}(10)$ Geschwindigkeitsgewinn.

geringem Berechnungsaufwand analytisch herleiten. Bei triangulierten Gittergeometrien müssen hingegen alle Eckpunkte gegeneinander getestet werden. Dieses Tests können in Abhängigkeit der Anzahl an beteiligten Dreiecken sehr aufwendig sein, selbst wenn dieses Verfahren parallelisiert umgesetzt würde.

Eine unter Umständen effizientere Methode für bewegliche und nicht verformbare Körper ist, während der Initialisierung einmalig eine Kugel als Hüllkörpergeometrie der triangulierten Gittergeometrie zu bestimmen, deren Mittelpunkt mit dem Schwerpunkt des Körpers übereinstimmt, vgl. Abb. 28.

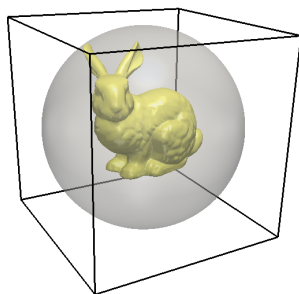


Abbildung 28: Vereinfachter Hüllkörper einer triangulierten Gittergeometrie. AABB der Hüllkugel, dessen Mittelpunkt mit dem Schwerpunkt der interessierenden Körpers übereinstimmt am Beispiel des *Stanford Bunny* [57].

Anschließend wird der Radius der Kugel bestimmt und erlaubt jederzeit ein einfaches Rekonstruieren einer validen Hüllkörperkugel, selbst wenn der Körper um seinen Schwerpunkt verdreht ist. Zur einfacheren Umrechnung auf Gitterkoordinaten wird von dieser Hüllkörperkugel eine AABB bestimmt. Um den Neuberechnungsaufwand zu vermeiden, wird diese für die weitere Simulation abgespeichert. Ein Nachteil ist, dass diese AABB im Allgemeinen größer als die echte AABB der triangulierten Gittergeometrie ist, was in den meisten Fällen vernachlässigt werden kann. Nach

Bestimmung der AABB der Hüllkörperkugel werden die Methoden der Haftrandbedingung und die der Kraftberechnung auf die in ihr befindlichen Gitterknoten angewandt.

Je nach Anzahl und Größe der beteiligten Strömungshindernisse sowie der Anzahl ihrer Dreiecke, ließen sich mit beiden Techniken Simulationen um die Größenordnung $\mathcal{O}(10)$ beschleunigen.

6.3 Parallelisierte Addition

Die nach Kapitel 5.1 berechneten Teilfluidlasten lassen sich sehr effizient zu einer Gesamtfluidlast aufsummieren, die auf ein Strömungshindernis wirkt. Das Addieren vieler Summanden ist prädestiniert für eine hocheffiziente Parallelisierung auf Grafikkarten nach dem sog. *teile und herrsche*-Prinzip (engl. *divide and conquer*), vgl. Abb. 29.

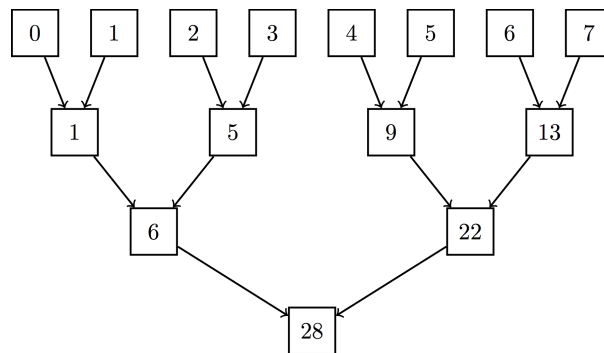


Abbildung 29: Vereinfachte Darstellung einer parallelisierten Addition. Jede Teilsumme wird durch einen *Thread* berechnet. In jeder tiefer liegenden Ebene, der sogenannten *stage*, halbiert sich die Anzahl der noch bestehenden Teilsummen, bis nur noch eine Endsumme übrig bleibt.

Zwar gibt es eine Vielzahl an Bibliotheken, wie z.B. CUB [70], die parallelisierte Reduktionsalgorithmen, wie bspw. `sum()`, `min()` und `avg()`, zur Verfügung stellen, jedoch setzen viele eine bestimmte Datenstruktur, einen bestimmten Speicherort, eine zur Kompilierungszeit feste Größe des Datensatzes voraus oder lassen sich nicht an beliebiger Stelle im Quellcode ohne umfangreiche Vorinitialisierung nutzen.

Inspiziert durch die Beispiele von Harris [45] wurden verschiedene geeignetere Reduktionsalgorithmen zur parallelen Aufsummierung erfolgreich entwickelt und implementiert. Sie gestatten die hocheffiziente Summierung beliebiger Datentypen sowie einer beliebigen Datensatzgröße n und sind von der Ordnung $\mathcal{O}(\log n)$. Sie arbeiten nativ auf derselben Speicherarchitektur, ohne diese zu verlassen und nutzen low-level Methoden der Grafikkartenarchitektur aus. Im Vergleich zu einer sequentiellen Addition gestatten die neu implementierten Methoden eine höhere GPU-Auslastung, sind wesentlich effizienter und beschleunigen die Berechnung der Fluidlasten deutlich. Die gesamte Kraftberechnungsmethode, inkl. der Teillastbestimmung nach Kapitel 5.1, ließ sich daher um den Faktor ≈ 6 beschleunigen.

6.4 stream-Parallelisierung

Während einer Mehrkörpersimulation werden unterschiedliche LBM-Methoden auf Gitterknoten ausgeführt, die sich im und um ein Strömungshindernis befinden, beispielsweise das Vergittern, das Reinitialisieren, die Haftrandbedingung oder die Kraftberechnungsmethode. Jede dieser Methoden muss auf jedes Strömungshindernis angewandt werden, womit der Gesamtaufwand linear mit der Anzahl der Hindernisse steigt. Eine Optimierung und Beschleunigung der Algorithmen dieser Methoden ist zweifelsohne von Vorteil. Allerdings ist irgendwann der Punkt erreicht, an dem eine weitere Minimierung kaum noch Vorteile bietet. Insbesondere tritt dies auf, wenn jede einzelne Methode für jedes Strömungshindernis auf nur noch einige wenige Gitterknoten gewandt wird. Eine parallele Problemlösung mittels Grafikkarten kann dann sogar ineffizienter werden. In der Regel tritt dies auf, wenn die Problemgröße unter $\mathcal{O}(10^6)$ fällt und somit die Grafikkarte nicht mehr vollständig ausgelastet werden kann. Um diesem Problem entgegenzuwirken, bietet die CUDA Entwicklungsumgebung eine Technik zum parallelen bzw. überlappenden Aufrufen der CUDA-Kernel mittels sogenannter **streams** an.

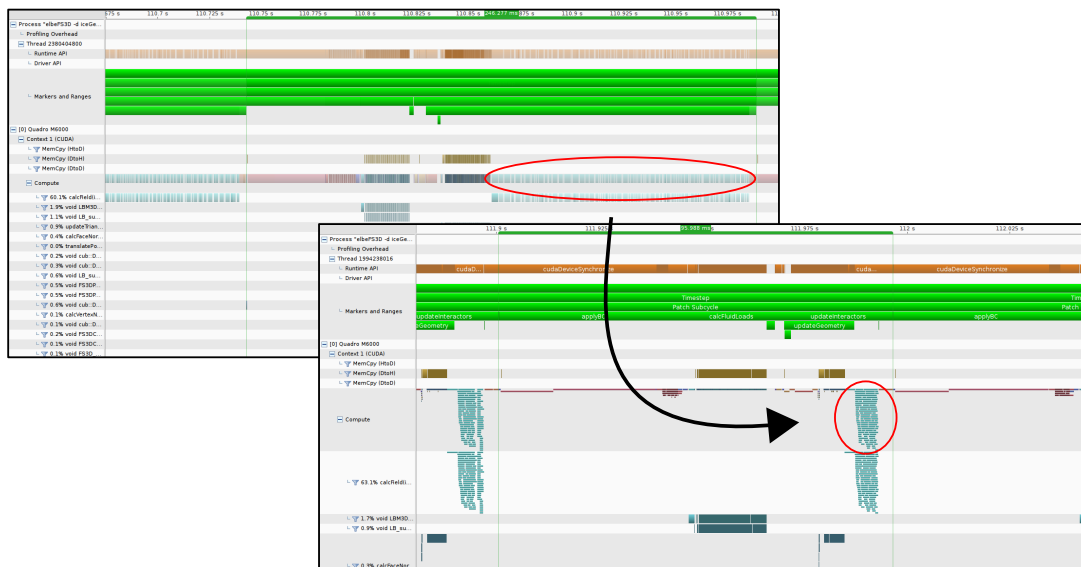


Abbildung 30: Kernel-Parallelisierung mittels **stream** am Beispiel der Vergitterungsmethode bei etwa 100 Eisschollen, eingekreist in rot. Mittels *Nvidia Visual Profiler* wird die sequentielle (oben) gegen den parallelisierten Kernelaufruf (unten) verglichen. Die Zeitlinie verläuft von links nach rechts. Dank **stream**-Parallelisierung konnte die Berechnungsdauer der Vergitterung um den Faktor ≈ 15 verkürzt werden.

In **streams** lassen sich einzelne Operationen auf der Grafikkarte, wie zum Beispiel ein Kernel-Aufruf oder das Kopieren von Daten von *Host* (CPU-betreffend) zu *Device* (GPU-betreffend), auslagern bzw. an einen **stream** binden. Mehrere **streams** lassen sich anschließend parallel bzw. überlappend ausführen. Dies ist insbesondere von Vorteil, wenn es viele Kernel-Aufrufe gibt, die nur eine geringe Menge an Threads benötigen. Eine sequentielle Bearbeitung dieser Kernel würde die Grafikkarte nicht voll auslasten können, eine parallele Bearbeitung hingegen schon.

Im Rahmen des Projekts wurde `elbe` umfangreich umgeschrieben, um Vorteile aus `streams` ziehen zu können. Für das Vergittern, die Randbedingungen sowie für die Kraftberechnung konnte eine erfolgreiche Parallelisierung mittels `streams` erzielt und die gesamte Effizienz dadurch deutlich gesteigert werden, vgl. Abb. 30.

In der Praxis zeigte sich eine Beschleunigung der Simulationen um den Faktor ≈ 3 .

7 Vereinfachtes Propulsionsmodell

In Kooperation mit dem Projektpartner HSVA wurde eine Software-Bibliothek zur Propulsionsmodellierung entwickelt. Die hierbei entstandene `LibPPB` dient zur Ermittlung der induzierten Strömungsgeschwindigkeiten eines Schraubenpropellers an beliebigen Punkten des Nachstroms. Die `LibPPB` wurde von der HSVA in der Programmiersprache `Fortran` geschrieben und als *closed-source* Software-Bibliothek übergeben. Sie unterstützt den instationären Zustrom zum Propulsor, Ein- und Zweischrauber sowie Drallentwicklung und ermöglicht die Modellierung von Schraubenpropellern in `elbe`, ohne diese aufwendig im Strömungsfeld geometrisch voll auflösen zu müssen. Das bedeutet, dass die Strömung den Propeller nicht direkt *sieht*. An der Kollisionsdynamik können die Propellergeometrien dennoch beteiligt sein.

Zur Anbindung der Bibliothek an `elbe` wurde eine monolithische und bidirektionale Kopplungsschnittstelle geschaffen, die für den Datenaustausch zwischen `elbe` und der `LibPPB` sorgt. Die Schnittstelle ermittelt zunächst den instationären Zustrom im Strömungsfeld auf einer Gitterknotenebene, die sich etwa $1D$ stromauf vom Propeller befindet, vgl. Abb. 31. Falls sich in dieser Messebene zur Zeit ein Strömungshindernis befindet und damit kein vollständiges Strömungsfeld vorliegt, werden die fehlenden strömungsrelevanten Daten aus den nächstgelegenen Fluidnachbarknoten mit nullter Ordnung extrapoliert. Anschließend wird der instationäre Zustrom als auch die Propellerbetriebsdaten an die `LibPPB` übermittelt. Daraufhin berechnet diese auf Basis eines Panelverfahrens die induzierten Strömungsgeschwindigkeiten um den Propeller. Diese werden von der Schnittstelle aufbereitet und etwa $0,5D$ vor der Propellerebene wahlweise als Kraftdichten oder als Geschwindigkeitsdifferenzen im Momentenraum auf das Strömungsfeld aufgebracht.

Im Falle von Kraftdichten wird die induzierte Geschwindigkeit $\Delta\mathbf{v}$ zunächst zu der Geschwindigkeit in der Messebene \mathbf{v}_{mess} addiert und beschreibt die *soll*-Geschwindigkeit \mathbf{v}_{soll} . Anschließend wird von dieser die bereits vorherrschende Geschwindigkeit in der Propellerebene \mathbf{v}_{ist} abgezogen und stellt damit die eigentliche induzierte Geschwindigkeit $\widetilde{\Delta\mathbf{v}}$ dar

$$\mathbf{v}_{\text{soll}} = \mathbf{v}_{\text{mess}} + \Delta\mathbf{v} \quad (40)$$

$$\widetilde{\Delta\mathbf{v}} = \mathbf{v}_{\text{soll}} - \mathbf{v}_{\text{ist}}, \quad (41)$$

welche daraufhin im gleichen Verfahren wie die Gravitation auf die Verteilungsfunktionen aufgebracht wird

$$f_i = f_i + 3w_i\mathbf{e}_i\widetilde{\Delta\mathbf{v}} \quad \forall i. \quad (42)$$

Für den Fall, dass die induzierten Geschwindigkeit $\Delta\mathbf{v}$ im Momentenraum aufgebracht werden sollen, wird der Geschwindigkeitsanteil der Momente der Propellerebene lediglich durch die Summe aus der Geschwindigkeit der Messebene \mathbf{v}_{mess} sowie der induzierten Geschwindigkeit $\Delta\mathbf{v}$

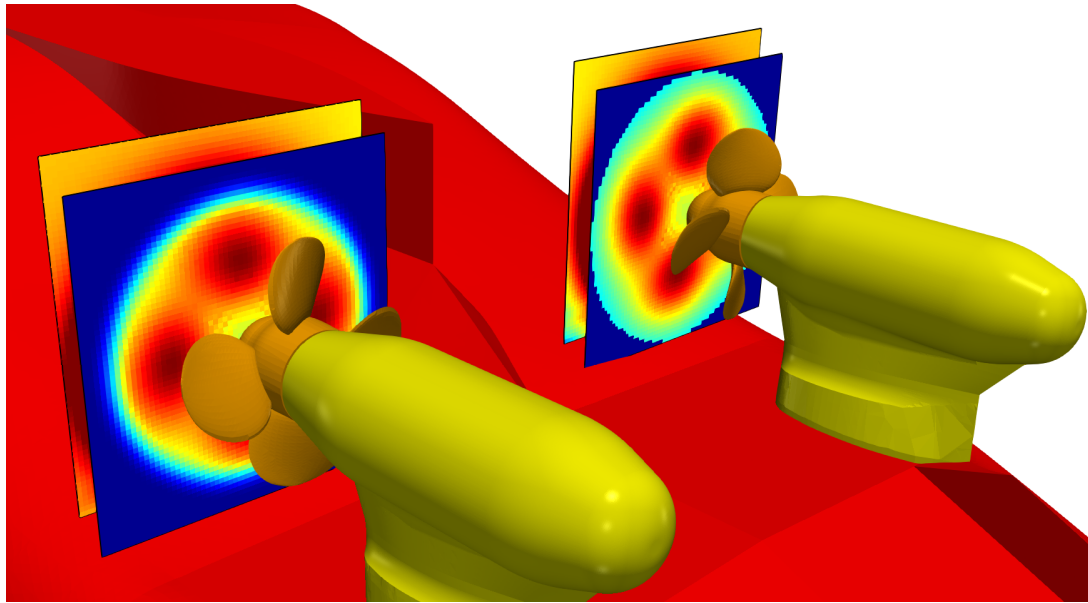


Abbildung 31: Anwendung des vereinfachten Propulsionsmodells LibPPB. Zu sehen sind zwei stromauf gerichtete Pod-Antriebe sowie die vor ihnen befindlichen Zustromfelder (teilweise verdeckt) und die Felder der durch die LibPPB bestimmten induzierten Strömungsgeschwindigkeiten. Zusätzlich ist auf der rechten Seite zur Veranschaulichung die polare Rampenfunktion deaktiviert und somit die Unstetigkeit in der induzierten Geschwindigkeit ersichtlich.

ersetzt

$$\mathbf{v}_{\text{ist}} \equiv \mathbf{v}_{\text{mess}} + \Delta \mathbf{v}. \quad (43)$$

Falls die durch die LibPPB bestimmten induzierten Geschwindigkeiten starke Ausreißer aufweisen sollten, dann lässt sich optional ein Median Filter zur Korrektur hinzuschalten. Dies war bisher jedoch nur in wenigen Fällen notwendig. Des Weiteren wird aufgrund einer Unstetigkeit der induzierten Geschwindigkeiten im Bereich der Rechengebietsgrenze der LibPPB mittels einer polaren Rampenfunktion die induzierte Geschwindigkeit im Fernfeld des Propellers auf Null minimiert, vergleiche linken und rechten Propeller in Abb. 31.

8 Eisschollengenerator

Um eine beliebig lange Interaktion zwischen Schiff, Fluid und Eis simulieren zu können, bedarf es einer kontinuierlichen Zu-, als auch Abfuhr der Eisschollen in das Rechengebiet hinein bzw. heraus. Im Vergleich zu einer Vorinitialisierung eines endlichen Eisschollenfeldes inklusive Vorlaufzeit ermöglicht ein solcher *Eisschollengenerator* eine signifikante Verkleinerung des Rechengebiets und gestattet damit direkt eine feinere räumliche Auflösung. Hierfür wurden zwei Konzepte entwickelt.

Konzept *eins* basiert darauf, die neu hinzuzuführenden Eisschollen (inkl. Geometrie und Randbedingung) während der Programmlaufzeit vollkommen dynamisch zu erstellen und zu einem späteren Zeitpunkt – wenn sie das Rechengebiet verlassen – zu zerstören bzw. den Speicherplatz freizugeben. Dieses Konzept bedarf umfangreicher Umbaumaßnahmen an `elbe`. Insbesondere ist darauf zu achten, dass der allozierte Speicher nach Gebrauch einer Eisscholle komplett freigegeben wird und nichts verloren geht, da ansonsten Speicherlecks auftreten können, die zu einer kontinuierlichen Erhöhung der Speicherplatzausnutzung bis hin zum Programmabsturz führen können. Außerdem kann unter Umständen die Erstellung einer neuen Eisscholle relativ ineffizient sein. Ein Vorteil ist, dass problemlos beliebig viele Eisschollen erstellt werden und zeitgleich existieren können.

Konzept *zwei* sieht vor, zum Programmstart eine bestimmte Menge an pre-initialisierten Eisschollen vorzuhalten, die bei Nichtnutzung in einem imaginären Pool zwischengespeichert und zur Laufzeit entnommen und dem Rechengebiet zugeführt werden können. Das spart die unter Umständen umfangreiche Erstellung der Eisschollen zur Programmlaufzeit. Der Nachteil besteht darin, dass bereits zum Programmstart eine ausreichend hohe Anzahl an Eisschollen definiert werden muss. Zu Engpässen kann es während der Laufzeit kommen, wenn sich bereits alle Eisschollen im Rechengebiet befinden (und der Pool somit leer ist) und weniger Eisschollen das Rechengebiet verlassen als zugeführt werden müssen. In diesem Szenario kann erst eine Eisscholle dem Rechengebiet zugeführt werden, wenn eine andere das Rechengebiet verlässt. Eine Abschätzung der benötigten Eisschollen ist daher sehr schwierig und abhängig von dem jeweils gewählten Testszenario. Außerdem ist es im Falle von vordefinierten Eisschollenformen (den sogenannten *presawn*-Eisschollen) nicht trivial möglich eine austretende an der Position einer eintretenden Eisscholle zu setzen, da unter Umständen für diese Position eine andere Eisschollenform vorgesehen ist.

Da beide Konzepte sowohl Vor- als auch Nachteile bieten und sich keines favorisieren ließ, wurden zunächst beide Konzepte in einer Testphase implementiert. Nach ausreichendem Fortschritt kristallisierte sich Konzept *eins* als das vielversprechendere Konzept heraus, da sich die Implementierung einer dynamischen Speicherverwaltung einfacher herausstellte als zunächst vermutet. Zur Erfassung des aktuellen Status einer Eisscholle werden Hüllkörper herangezogen, welche die verschiedenen Zonen definieren, vgl. Abb. 32. Die Initialisierungszone befindet sich außerhalb stromauf vor dem Rechengebiet. Hier werden neue Eisschollen störungsfrei hinzugefügt und

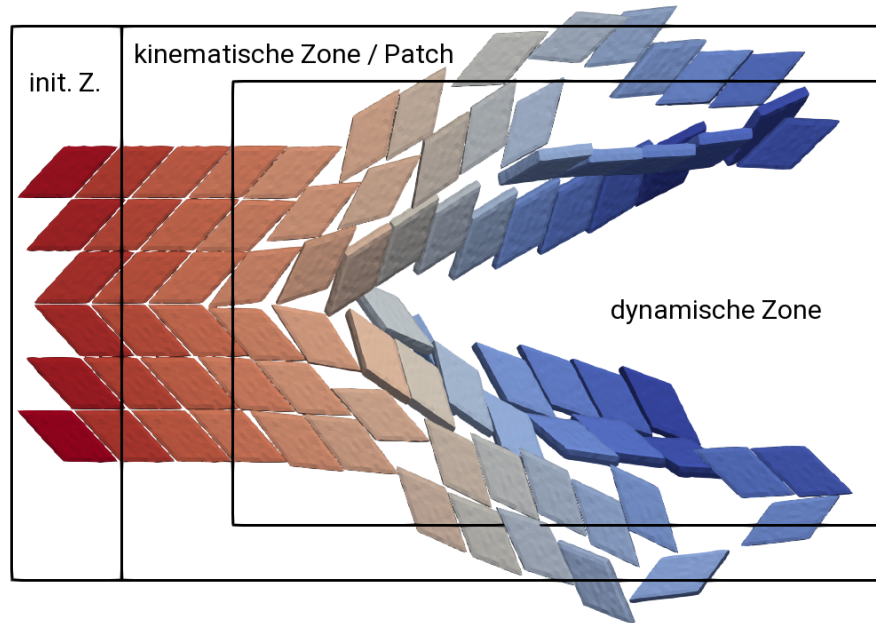


Abbildung 32: Zufuhr von Eisschollen mit vordefiniertem Schnittmuster. Abgebildet sind die Eisschollen (drei Reihen pro Schiffseite), die um einen (hier ausgeblendeten) Schiffsrumpf transportiert werden, inkl. der Initialisierungszone, der kinematischen Zone und der dynamischen Zone.

vollführen keinerlei Interaktion untereinander. Die kinematische Zone beschreibt den Raum um den inneren Rand des Strömungsgebietes und beinhaltet außerdem die Initialisierungszone. Die Breite dieser Zone entspricht der maximalen Kantenlänge der verwendeten Eisschollengeometrie. Eisschollen, die sich in dieser Zone befinden, werden mit konstanter Geschwindigkeit kinematisch geführt und können sowohl auf dynamisch geführte Eisschollen als auch auf das Fluid einwirken. Dies sorgt für eine störungsfreie Zu- und Abfuhr der Eisschollen. Auf Eisschollen, die sich nur teilweise im Strömungsgebiet befinden, wirkt nur eine anteilige Auftriebskraft, die zu einem physikalisch inkonsistenten Bewegungsverhalten führen würde. Daher werden die Eisschollen außerhalb des Strömungsgebiets initialisiert und anschließend kinematisch in das Strömungsgebiet eingeführt, sodass sich mit dem Fortschreiten ein Gleichgewicht zwischen Auftriebs- und Gewichtskraft der Körper einstellen kann. Sobald eine Eisscholle die kinematische Zone passiert hat, gelangt sie in die dynamische Zone. In dieser Zone reagiert die Eisscholle dynamisch und kann mit dem Fluid, den anderen Eisschollen, dem Schiffsrumpf sowie den Propulsionsorganen interagieren, bis sie in die seitliche, obere oder hintere kinematische Zone gelangt und kinematisch aus dem Rechengebiet abgeführt wird. Sobald diese die kinematische Zone verlässt, wird die Eisscholle zerstört und der Speicherplatz wird freigegeben.

Bei der Zufuhr von Eisschollen stehen zwei verschiedene Anordnungsarten zur Auswahl. Zum einen kann ein Teppich an Eisschollen nach einem vom Projektpartner HSVA vordefinierten Schnitt-/Bruchmuster zugeführt werden, mit dem sich eine Eisbedeckung von nahezu 100% erzielen lässt, vgl. Abb. 32. Zum anderen besteht die Möglichkeit der dynamischen Zufuhr von

Eisschollen beliebiger Form und Größe, um einen spezifischen Eisbedeckungsgrad zu erzielen. Dies wird erreicht, indem stetig zufällige Eisschollen hinzugeführt werden, bis der gewünschte Bedeckungsgrad erreicht ist. Bei der Wahl der Initialisierungsposition ist hierbei darauf zu achten, dass die potentiell neu hinzugeführte Eisscholle nicht mit den bereits vorhandenen Körpern kollidiert. Dieser Punkt lässt sich mittels der Kollisionsdetektion der *Open Dynamics Engine* leicht bestimmen. Ist dies der Fall, wird die Eisscholle entweder neu positioniert, rotiert oder verkleinert bis der gewünschte Bedeckungsgrad erlangt ist.

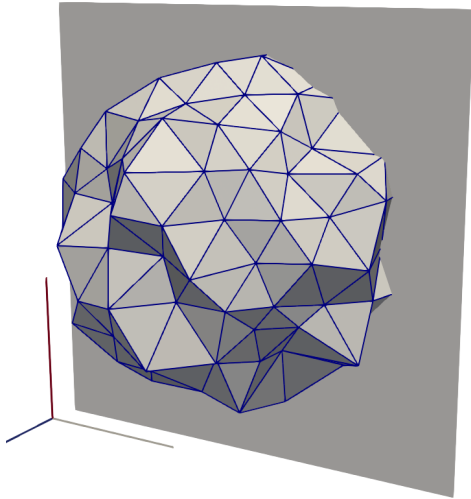
Des Weiteren wurde eine Methode zur Bestimmung des Eisbedeckungsgrades implementiert. Um einen möglichst ungestörten bzw. Fernfeld-Bedeckungsgrad bestimmen zu können, wird der Bereich der vorderen kinematischen Zone als Messzone definiert. Zur Messung bieten sich grundlegend zwei Techniken an. Eine Möglichkeit stellt das Zählen der Gitterknoten, die sich in der x - y -Ebene auf Höhe der Eisschollen befinden und als Hindernisknoten markiert sind, dar. Diese Technik ist relativ einfach, allerdings unpräzise, insbesondere weil ihre Genauigkeit von der Gitterauflösung abhängt und die geometrische Abdeckung nur indirekt auf Basis der vergitterten Knoten gemessen wird. Eine Technik mit variabler Genauigkeit bietet die Messung mittels Strahlen. Hierbei wird auf die Kollisionsdetektion der *Open Dynamics Engine* zurückgegriffen, indem virtuelle Strahlen auf der Messzone verteilt werden, die parallel zur z -Achse gerichtet sind. Anschließend wird die Anzahl an Kollisionen zwischen den Strahlen mit den Eisschollen bestimmt. In Relation zur Anzahl der Strahlen beschreibt diese Größe den Bedeckungsgrad. Um die Bedeckung ausreichen genau messen zu können, sind mindestens $\mathcal{O}(100)$ Strahlen erforderlich. Der hierfür benötigte Berechnungsaufwand fällt relativ gering aus.

9 Messmethode zur Analyse des Eisdurchflusses in der Propellerebene

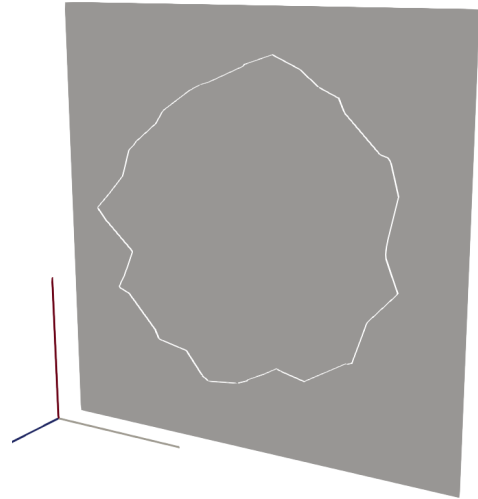
Zur Quantifizierung des zeitlichen Eis-Volumenstroms in der Propellerebene wurde auf Basis der Anforderungen seitens des Projektpartners HSVA eine effiziente und genaue Messmethode entwickelt. Die Anforderungen legen hierfür fest, dass Eisschollengeometrien, die sich in der Messebene befinden, jeweils zu einem flächenerhaltenden Rechteck diskretisiert werden, dessen Eckkoordinaten in einer Datei gespeichert werden.

Um eine Überschneidung zwischen der Eisscholle und der Messebene effizient zu ermitteln, wird zunächst getestet, ob der Hüllkörper der Eisscholle von der Messebene geschnitten wird. Ist dies der Fall, wird weiterhin getestet, ob eine echte Überschneidung zwischen der Geometrie und der Messebene vorliegt. Wenn auch diese Bedingung erfüllt ist, wird die Schnittmenge generiert, welche einem geschlossenen Polygonzug entspricht. Je nach Form der Geometrie können sich pro Schnittmenge auch mehrere Polygonzüge ergeben, die zu identifizieren und zu isolieren sind. Anschließend wird von jedem Polygon die Fläche, der Flächenschwerpunkt und der dazugehörige konvexe Polygonzug bestimmt. Zu diesem wird daraufhin der kleinstmögliche orientierte Hüllkörper generiert und anschließend auf die Fläche des originalen Polygons zurück skaliert. Das somit resultierende, durch vier Punkte diskretisierte flächenkongruente Polygon entspricht den Anforderungen der HSVA und wird anschließend inklusive Zeitstempel sowie Eisschollen- und Sub-Polygonindex fortlaufend in einer Datei abgespeichert. Eine Zusammenfassung der Verfahrensschritte anhand von Graphiken findet sich in Abb. 33.

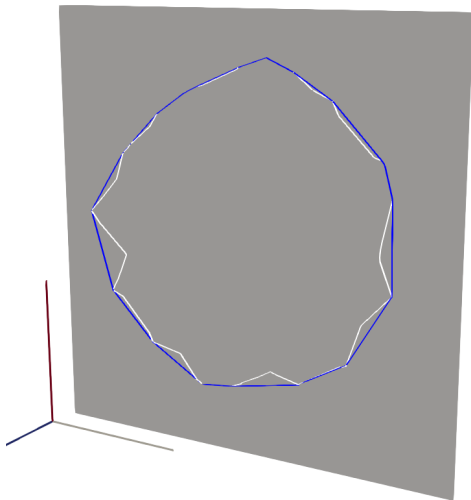
Das gesamte Verfahren ist nicht trivial. Insbesondere das Generieren einer polygonalen Schnittmenge zwischen Eisschollengeometrie und Messebene gestaltete sich als äußerst umfangreich und komplex. Nach der Implementierung sowie der Tests diverser Ansätze wurde letztendlich auf eine externe Methode aus der VTK-Bibliothek [85] zurückgegriffen. Insgesamt erweist sich die Messmethode als ausreichend effizient und äußerst genau.



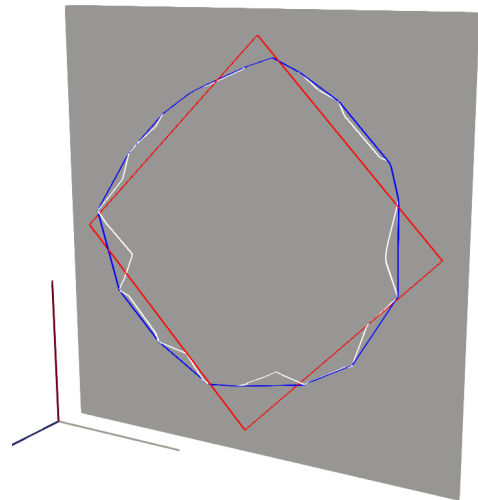
(a) Geometrie und Messebene überlappen sich.



(b) Schnittmenge zwischen der Geometrie und der Messebene als Polygonzug.



(c) Konvexe Polygonzug der Schnittmenge.



(d) Flächenkongruentes Rechteck des kleinstmöglichen orientierten Hüllkörpers des konvexen Polygons.

Abbildung 33: Methode zur Analyse der Propellerdurchströmung. Verfahrensschritte zur Diskretisierung der Schnittmenge zwischen einer Geometrie und einer Messebene.

10 Algorithmus zur Neuvergitterung aufbrechender Geometrien

Innerhalb des Forschungsprojektes wurde ein Algorithmus zur Neuvergitterung von aufbrechenden Geometrien entwickelt. Dieser kann genutzt werden, um Eisschollen, die durch Propellerkollision zerteilt werden, an den entstehenden Schnittebenen neu zu vernetzen. Hierfür wurde ein schneller und stabiler Algorithmus geschaffen, der im Folgenden erläutert wird.

Zur einfachen und gleichzeitig effizienten Modifizierung und Neuvergitterung von triangulierten Gittergeometrien wurde auf die bestehende *OpenMesh*-Bibliothek [77] zurückgegriffen. Diese wurde um viele verschiedene Methoden umfangreich erweitert. *OpenMesh* dient zur Darstellung und Manipulation von polygonalen Netzen und basiert auf der sogenannten Halbkanten-Datenstruktur (engl. *half-edge data structure*). Diese sieht vor, dass Kanten von polygonalen Netzen durch jeweils zwei Halbkanten beschrieben werden. Der eigentliche Vorteil dieser Datenstruktur besteht darin, dass von einer Halbkante aus betrachtet alle nächstgelegenen topologischen Objekte mit konstantem Aufwand erreicht werden können. Das heißt, dass es nur einen *Sprung* benötigt, um von einer Halbkante zur nächsten bzw. vorherigen Halbkante oder Eckpunkt als auch zur angrenzenden oder gegenüberliegenden Fläche zu gelangen. Somit lassen sich Nachbarschaftsfragen schnell beantworten, effiziente Iterationen um Flächen und Eckpunkte – insbesondere auch auf unstrukturierten Gittern – tätigen und somit stabile Algorithmen zur Modifikation von Geometrien entwickeln.

Zur Neuvergitterung aufbrechender Geometrieflächen sind mehrere Teilschritte bzw. Operationen entwickelt worden:

- `convert()`: Konvertieren von und zur *OpenMesh* Halbkanten-Datenstruktur
- `knife()`: Schneiden und Neu-Triangulieren einer Geometrie entlang einer Schnittebene
- `split()`: Zertrennen der Ausgangsgeometrie entlang der Schnittebene in zwei neue Geometrien
- `findHole()`: Lokalisieren der Öffnungen beider neuen Geometrien
- `fillHole()`: Schließen der Öffnungen beider Geometrien
- `getVolumeProperties()`: Bestimmung der neuen Geometrieigenschaften (Volumen, Volumenschwerpunkt, Trägheitsmoment)

Insbesondere die `fillHole()`-Operation basiert auf einem sehr umfangreichen Algorithmus. Dieser sieht vor, möglichst gleichseitige Dreiecke zum Schließen der Öffnung zu verwenden, die im Vergitterungsverfahren wesentlich vorteilhafter sind. Hierzu wurde ein Ansatz auf Basis der *Advancing Front Method* [16] inklusive des *Delaunay*-Triangulationskriteriums [59] gewählt. Diese Methode besteht daraus, von der offenen Berandung ausgehend an der Stelle mit dem kleinsten Öffnungswinkel sukzessiv Dreiecke hinzuzufügen, bis die Öffnung komplett geschlossen ist, vgl. Abb. 34.

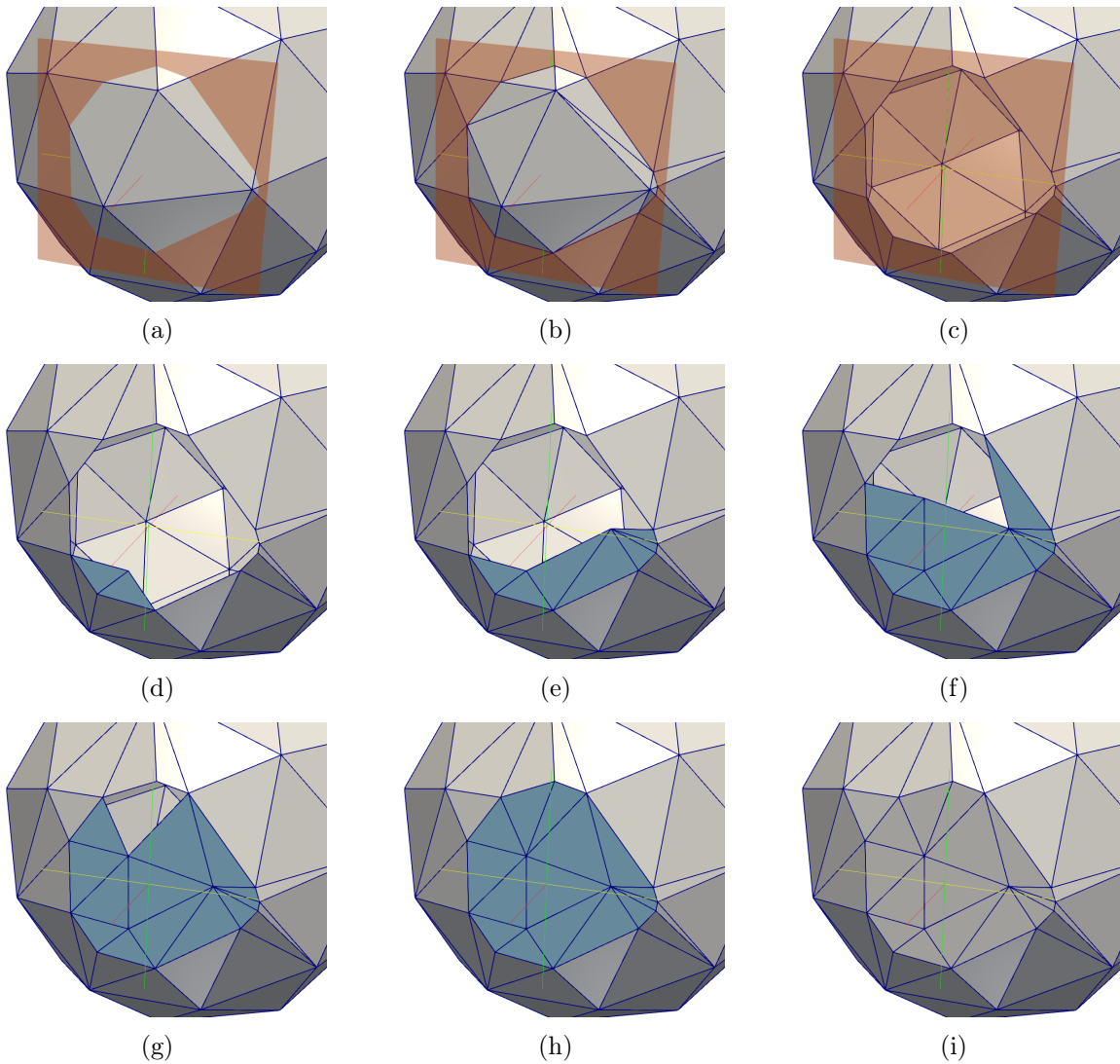


Abbildung 34: Algorithmus zur Neuvergitterung aufbrechender Geometrien. Zu sehen ist eine Ausgangsgeometrie und eine Schnittebene (a), deren Schnittmenge mittels `knife()`-Operation zusätzliche Dreiecke zur Geometrien hinzufügt (b). Anschließend werden mittels `split()`-Operation die Geometrien voneinander getrennt (c). Nachdem mit der `findHole()`-Operation die Öffnung gefunden wurde, wird/werden mittels `fillHole()` stets an der Stelle mit dem kleinsten Öffnungswinkel, abhängig von der Größe des Öffnungswinkels ein, zwei oder drei Dreiecke hinzugefügt (d)-(g). Sobald die Öffnung geschlossen ist (h), wird mittels `getVolumeProperties()` Schwerpunkt, Volumen und Trägheit bestimmt (i).

Basierend auf dem Öffnungswinkel werden drei Fälle unterschieden: entweder werden ein ($< 75^\circ$), zwei ($< 100^\circ$) oder drei ($< 180^\circ$) Dreiecke je Schritt hinzugefügt. Die Methode bietet zahlreiche Parameter, mittels derer die resultierende Gitterqualität stark beeinflussen werden kann und zu denen ein optimaler Parametersatz identifiziert werden konnte.

Der erfolgreich entwickelte Algorithmus zur Neuvergitterung von aufbrechenden Geometrien konnte leider keine Anwendung für die final zur Validierung des Simulationstools herangezogenen komplexen Testfälle – vgl. Kapitel 13 – im Forschungsprojekt finden. Dies lag zum Teil

daran, dass sich mit Fortschreiten des Projektes das Interesse der Projektpartner von ehemals schneidenden auf spanende Trennmethode verlagerte. Eine numerische Modellierung von zer-spanenden Trennmethode liegt aufgrund der sehr hohen Anzahl von neu entstehenden Körpern in Bezug auf das Vergitterungsverfahren und die Effizienz der Gesamtroutine einige Größenordnungen über dem momentan numerisch umsetzbaren Rahmen. Die Weiterentwicklung eines Konzeptes zur Definition der Trennkraft, die zur Schubreduktion führt, wurde daher nicht weiter verfolgt.

11 Haftrandbedingung höherer Ordnung

Eine innovative Methode zur effizienten, stabilen und genauen sowie speicherarmen Bestimmung der normalisierten Zwischengitterabstände von Strömungshindernissen konnte erfolgreich entwickelt werden. Die Methode wurde auf der *International Conference for Mesoscopic Methods in Engineering and Science* (ICMMES 2016) Konferenz vorgestellt und später in einem Journal veröffentlicht, vgl. [75].

Die normierten Zwischengitterabstände gestatten die Verwendung von Haftrandbedingungen höherer Ordnung, mit denen sich Strömungsprobleme untersuchen lassen, die mit bisherigen `e1be`-Methoden nur bedingt zu lösen waren. Insbesondere bei Haftrandbedingungen, die bewegten Strömungshindernissen, wie etwa Eisschollen, mitgeführt werden, zeigte sich der elementare Vorteil solcher Methoden höherer Ordnung. Sie ermöglichen eine bedeutend genauere Berücksichtigung der diskreten geometrischen Formgebung der Strömungshindernisse und erlauben wesentlich störungsärmere Simulationen durch eine divergenzfreie Initialisierung neuer Fluidknoten. Ebenso eignet sich die genauere Berücksichtigung der Formgebung für nicht bewegliche und grob diskretisierte Strömungshindernisse, wie zum Beispiel den Azi-Pod Gondeln, vgl. Kapitel 13.1. Das Hauptmerkmal der neu entwickelten Methode liegt darin, dass sie komplett auf der Grafikkarte ausgeführt wird und die erforderlichen Eingangsparameter als Nebenprodukte bestehender Algorithmen anfallen, wodurch diese kompakt zwischengespeichert werden können. Als Zwischenschritt diskretisiert die Methode die Geometrie der Strömungshindernisse auf die LBM-Gitterknoten und rekonstruiert die normalisierten Zwischengitterabstände aus dieser charakteristischen Formgebung.

Eine ausführliche Fehlerkonvergenzstudie zeigte, dass die Genauigkeit der Methode direkt vom Auflösungsverhältnis der diskreten Geometrie zur Rechengitterweite (räumlichen Diskretisierung) abhängt. Im Falle einer diskreten Kugel- bzw. Kreisgeometrie konnte festgestellt werden, dass eine höchstmögliche Genauigkeit erzielt werden kann, wenn die mit der Gitterweite normierte charakteristische Geometriesegmentlänge etwa 0.5 entspricht. Die Methode wurde umfangreich optimiert und validiert und um eine Technik zur Reduktion des zusätzlichen Speicherbedarfs erweitert. Die Erweiterung ermöglicht einen zusätzlichen Rückgang des Speicherbedarfs um wahlweise den Faktor 2 oder 4, bei nahezu gleichbleibender Berechnungsqualität. Ausführlicher wird die Methode in [75] präsentiert.

12 Anpassungen an die Simulation eines Voith-Schneider-Propellers (VSP)

In diesem Abschnitt werden die Anpassungen des Strömungslösers erläutert, welche aus den Anforderungen an die Simulation eines Voith-Schneider-Propellers (VSP) resultieren. Die Implementierung der VSP-Kinematik wurde in enger Abstimmung mit dem Projektpartner Voith durchgeführt. Hierzu wurden sowohl Konfigurationen mit einem als auch mit zwei VSP Radkörpern – ein Radkörper stellt hierbei einen separaten Drehkreis bestehend aus fünf VSP-Profilen dar – vorgesehen. Eine detaillierte Beschreibung der Validierung der implementierten VSP-Kinematik erfolgt in Kapitel 13.2.

Die in Kapitel 2.6 entwickelten zonalen Methoden dienen der lokalen Gitterverfeinerung im Bereich der freien Wasseroberfläche, der Eisschollen und des Propellers. Während diese drei Verfeinerungs-Zonen in konventionellen Konfigurationen (Schraubenpropeller oder Azi-Pod-Antrieb) nahezu übereinstimmen und lediglich eine Gitterverfeinerungszone (parallel zur Wasseroberfläche ausgerichtet) notwendig ist, ist dieses bei der Verwendung von Zyklodialpropellern nicht der Fall. Die nahezu rechtwinklig zur Wasseroberfläche ausgerichtete Propellerachse erfordert zudem eine angepasste, entsprechend ausgerichtete Gitterverfeinerung. Dies führt dazu, dass die untersuchten Eisschollen nach initialer, schwimmender Zufuhr an oder nahe der Wasseroberfläche nach dem Entlanggleiten am Schiffsrumpf in feiner vernetzte Gebiete eingesogen werden, vgl. Skizze in Abb. 35 (**Prop.-II**).

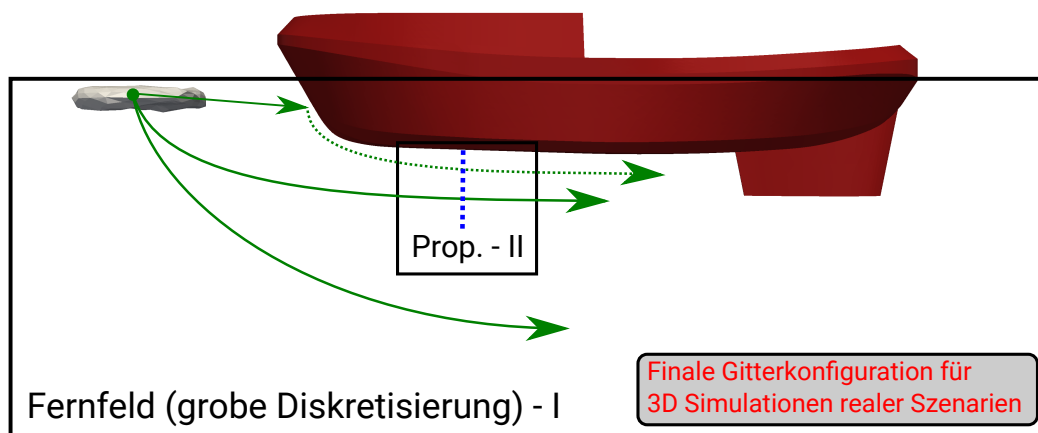


Abbildung 35: Gitterkonfiguration für den Voith-Schneider-Propeller Testfall. Nach der initialen Zufuhr der Eisschollen an oder in der Nähe der Wasseroberfläche in dem grob diskretisierten Fernfeld (I) gleiten diese am Rumpf entlang und werden anschließend oder auch direkt in die den Propulsor umgebende feiner diskretisierte Region (II) eingesogen. Die vertikal ausgerichtete Drehachse des VSP-Radkörpers ist blau gepunktet hervorgehoben.

Die Simulation eines VSPs erfordert zahlreiche algorithmische und strukturelle Anpassungen des Lösers. Die Bewegung des Propellerblattes durch das Rechengebiet macht eine Knoten-Reinitialisierung erforderlich, d.h. Knoten die im vorherigen Zeitschritt dem Geometrieobjekt

zugeordnet waren, sind im aktuellen Zeitschritt der Fluidphase angehörig, vgl. Abb. 36. Für diese Konvertierung von Knotenstatus ist eine Reinitialisierung von LBM Freiwerten (Partikelverteilungsfunktionen) erforderlich. Die einfachste Methode stellt hierbei eine Mittelwertbildung der makroskopischen Dichte und der Geschwindigkeit aus den umgebenden validen direkten Fluidnachbarknoten dar. Die Initialisierung der neuen Fluidknoten erfolgt anschließend durch das Aufprägen von Gleichgewichts-Verteilungsfunktionen auf Basis der berechneten Mittelwerte. Weitere Algorithmen zur Erhöhung der Genauigkeit des Reinitialisierungsschrittes umfassen die Extrapolation aus weiter in dem Feld liegenden Fluidknoten (Berücksichtigung der nächsten und übernächsten Fluidnachbarn) sowie eine komplexere Formulierung der Extrapolation unter Berücksichtigung der Körpergeschwindigkeit (engl. *Velocity Constraint Extrapolation* [63, 106]).

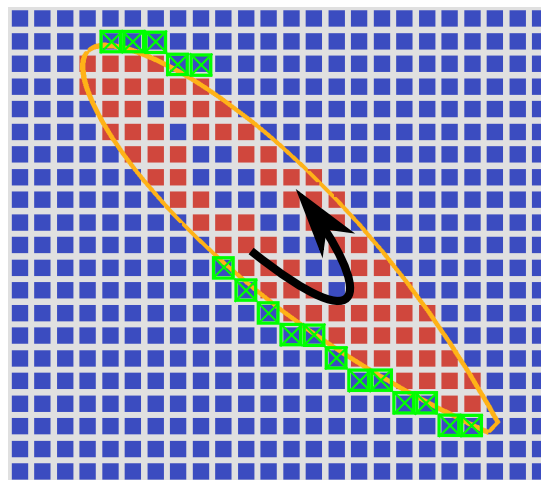


Abbildung 36: Skizze zur Problematik der erforderlichen Fluidknoten Reinitialisierung für bewegte Geometrieobjekte. Die Rotation des Profils (schwarzer Pfeil) erfordert nach der Neubestimmung der Knotenstatus eine Vorschrift zur Bestimmung valider Partikelverteilungsfunktionen auf den der Fluidphase neu zugeordneten Knoten (grün markiert).

Da die Partikelverteilungsfunktionen mittels der oben beschriebene Ansätze allesamt nicht divergenzfrei aufgebracht werden können, entstehen in Geometrienähe Fehlerterme, die einen negativen Einfluss in Form von Artefaken (sog. *Checkerboarding*-Effekt – (teils starke) Oszillationen des Geschwindigkeits- und Drucksignals) auf das gesamte Strömungsgebiet ausüben können, vgl. Abb. 37 (links, ohne Gitterverfeinerung. Die Entstehungszone und die Ausbreitung der Artefakte sind rot umrandet).

Um das Problem der Knoten-Reinitialisierung zu umgehen, wird der Ansatz körperfester mitbewegter Gitter verfolgt. Erste Entwicklungen im zweidimensionalen Raum wurden von [35] ausgeführt. Die umfassende Entwicklung sowie Implementierung in das 3D Einphasenmodul des Strömungslösers *e1be* ist in [13] beschrieben. Der Vorteil der Methode zeigt sich in Abb. 37 (rechts, körperfeste Gitterverfeinerung) in dem Strömungsfeld, das nicht von numerischen Artefakten überlagert wird. Hierbei bewegen sich die (verfeinerten) Gitter mit den Geometrien der VSP-Profilen. Statt einer Knoten-Reinitialisierung erfolgt in diesem Kontext eine (in 3D tri-

lineare) Interpolation (vgl. Kapitel 2.6.1) der LBM-Freiwerte in einem gewissen Abstand zur Geometrie im Übergangsbereich zwischen den Gittern. Diese Interpolation wurde umfangreich validiert und besitzt eine weitaus geringere Fehlerordnung als die verfügbaren Methoden zur Knoten-Reinitialisierung.

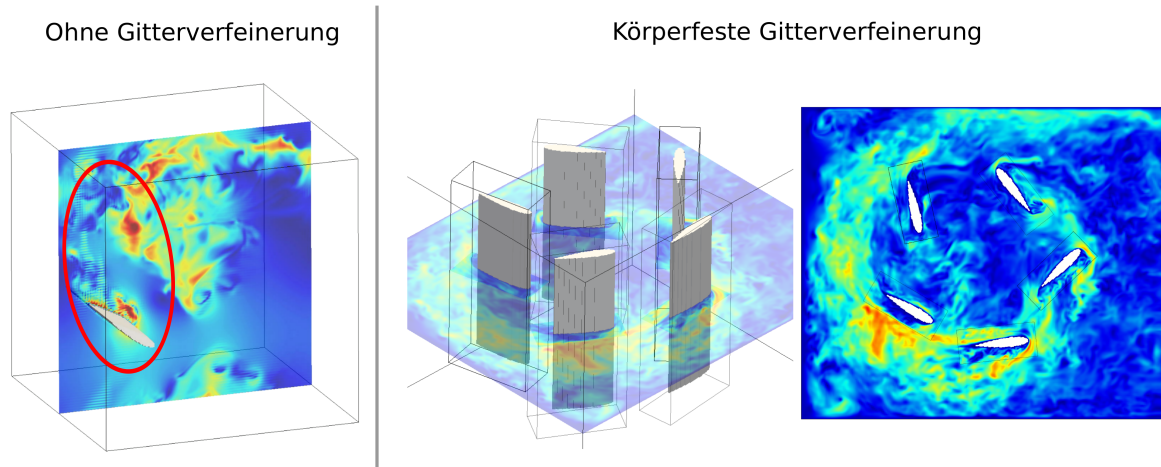


Abbildung 37: Bewegung von VSP-Profilen durch ein uniform diskretisiertes Rechengebiet ohne Gitterverfeinerung, das eine Knoten-Reinitialisierung erforderlich macht (links), die mit dem Aufprägen von Fehlerartefakten (rote Umrandung) einhergehen können. VSP-Radkörper mit fünf Flügeln, die jeweils mit einem lokalen mitbewegten Gitter (schwarze Boxen) versehen sind und zu einem nicht von numerischen Fehler beeinflussten Strömungsfeld führen.

Die neben der Interpolation der Verteilungsfunktionen verbleibende algorithmische Anpassung stellt die Behandlung von Starrkörperbewegungen über Gittergrenzen hinweg dar. Eisschollen, die auf den Rumpf zuströmen und entlang des Rumpfes abtauchen, wechseln von der relativ groben Diskretisierung des Fernfeldes (I) in die – die Propulsoren umgebende – Gitterverfeinerungsregion (II), vgl. Abb. 35.

Aufgrund der starken Kopplung der Raum- und Zeitdiskretisierung im Gitter-Boltzmann-Verfahren weist eine räumlich höher aufgelöste Gitterzone auch eine erhöhte zeitliche Auflösung auf. Die algorithmische Herausforderung besteht darin, dass sich Eisschollen gleichzeitig in Zonen unterschiedlicher Verfeinerungsstufe befinden können. Die unterschiedlichen Zeitschrittweiten der jeweiligen Zonen sind somit zudem in der Integration der auf die Schollen wirkenden Fluidlasten und bei der Aufprägung der Randbedingungen des Strömungsfeldes zu berücksichtigen. Darüber hinaus ist die Gitteraktualisierung nach der Schollenadvektion angepasst, um Inkonsistenzen zwischen den Gitter-Zonen zu vermeiden. Die Integration der Fluidlasten wurde hierzu über die in 5.1.3 beschriebene Methode auf Basis der Kraftberechnung über das Dreiecksnetz der zugrundeliegenden Dreiecksgeometrien gelöst, um eine stets eindeutige Zuordnung der Kraftbeiträge aus den jeweiligen Gittern zu gewährleisten.

Im Laufe der numerischen Umsetzung des finalen Anwendungsfalles zur Validierung des Lörsers anhand der im HSVA Modellversuch real gefahrenen Versuche hat sich herausgestellt, dass

die ursprünglich anvisierte, körperfeste Gitterverfeinerung je VSP-Profil für großskalige Anwendungsfälle – die mit einer hohen Schollenanzahl $\mathcal{O}(\geq 10^2)$ einhergehen (vgl. Kapitel 13) – mit einem deutlich zu hohen Effizienzverlust der Simulationen verbunden ist. Dies liegt unter anderem in den Überschneidungen der Schollen mit mehreren Verfeinerungszonen im Falle der Starrkörper-Kollisionen (Eisscholle-Eisscholle, Eisscholle-VSP-Profil) begründet.

Zudem handelt es sich bei der VSP-Simulation um ein klassisches Multiskalen Problem. Hierbei liegt die Umfangsgeschwindigkeit der VSP-Profile, aufgrund der hohen Drehrate, um mindestens eine Größenordnung höher als die der freien Anströmgeschwindigkeit. Zur Auflösung der involvierten Skalen sowie zur Gewährleistung der numerischen Stabilität des Lösers sind per se feine zeitliche wie räumliche Diskretisierungen zu verwenden, wodurch die auftretenden Reinitialisierungsfehler gering beziffert werden. Aufgrund des gewählten feinen expliziten Zeitschrittes der Simulation resultieren ohnehin lange Simulationszeiten, sodass ein weiterer signifikanter Effizienzverlust nicht tragbar ist.

Lediglich für sehr grobe Unterauflösungen in Raum und Zeit – die für die finalen Anwendungsfälle jedoch nicht vorliegen – ist die Methode der Knoten-Reinitialisierung nicht anwendbar. Grob diskretisierte Rechengebiete gehen mit signifikanten Gradienten der Strömungsgrößen zwischen den diskreten Knoten einher. Große Gradienten ziehen erhebliche Nicht-Gleichgewichtsanteile der um die neu zu setzenden Fluidknoten benachbarten validen Geber-Knoten nach sich, womit unter Verwendung der Reinitialisierung auf Basis gemittelter oder extrapolierter makroskopischen Größen und anschließender Verwendung von Gleichgewichtsverteilungen deutliche Fehlerterme in das Strömungsfeld aufgeprägt werden. Diese üben negative Einflüsse auf die Rechengenauigkeit sowie die numerische Stabilität der Simulation aus.

Die Umsetzung des finalen VSP-Testfalles greift auf eine den gesamten Propulsor umfassende Gitterverfeinerungszone zurück, vgl. Kapitel 13.2.1. Da diese Region alle VSP-Profile umfasst, ist folglich eine Knoten-Reinitialisierung vonnöten. Für die auszuführenden Parameterstudien wird zur Reduktion numerischer Artefakte die feinstmögliche Gitterdiskretisierung unter den gegebenen Speicherrestriktionen einer GPU herangezogen. Die aufgrund der auszuführenden Knoten-Reinitialisierungen auftretenden Oszillationen (geringe Amplitude) des hydrodynamischen Rohdatensignals werden im anschließenden Postprocessing unter Verwendung eines gleitenden Mittelwertfilters geglättet, vgl. Kapitel 13.2.3 und Kapitel 13.2.2.

13 Anwendung und Validierung des entwickelten numerischen Verfahrens

In diesem Kapitel wird die Validierung des im Rahmen des Forschungsprojektes umfangreich weiterentwickelten Strömungslösers **elbe** zur Simulation der Schiffs-Eis-Interaktion beschrieben. Die Validierung erfolgt hierbei anhand der Simulationen zweier komplexer anwendungsorientierter Modellversuchskonfigurationen grundlegend verschiedener Propulsionsarten. Einerseits wird ein mittels zweier Azi-Pod angetriebener Rumpf (Simulationen für den Projektpartner HSVA) numerisch untersucht und andererseits werden Modellversuchskonfigurationen von Rümpfen simuliert, die zum Antrieb auf den Voith-Schneider-Propeller zurückgreifen (Simulationen für den Projektpartner Voith). In Abb. 38 sind exemplarische Aufnahmen numerischer Simulationen für die beiden der Validierung zugrundeliegenden Propulsionsarten dargestellt.

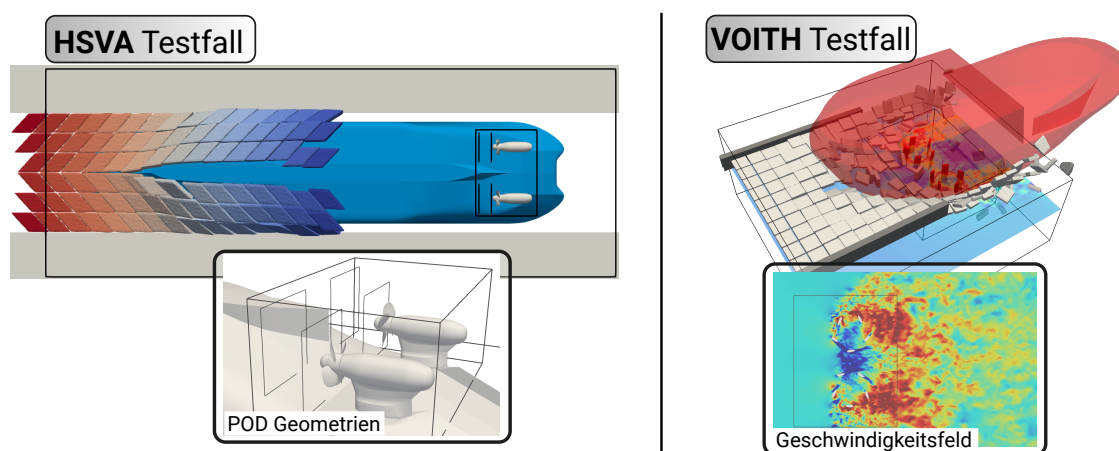


Abbildung 38: Für die Projektpartner HSVA und Voith ausgeführte Simulationen für zwei grundlegend verschiedene Propulsionsarten: Azi-Pod-Antrieb (links) sowie Voith-Schneider-Propeller (rechts).

Die Validierung beruht auf komplexen Testfällen, die möglichst alle im Zusammenhang mit PROEIS in den Löser implementierten Funktionalitäten nutzen. Das Ziel ist es die HSVA Konfigurationen der Modellversuche im Eistank möglichst detailgetreu numerisch nachzubilden und die Genauigkeit des Lösers sowie dessen Einsatzmöglichkeiten auf Grundlage quantitativer Größen (bspw. im Modellversuch gemessene Kräfte und Momente) oder auch qualitativer Studien (bspw. Analyse der Schollendynamik) beurteilen zu können.

Detaillierte Beschreibungen der Testfälle, eingeführte Modellbildungen sowie Auswertungen sind in den folgenden Kapiteln 13.1 und 13.2 zusammenfassend dargelegt.

13.1 Simulation des Eis-Volumenstroms am mo4977-Schiffsmodell (Azi-Pod Testfall)

Im Kooperationsvorhaben wurde zusammen mit dem Projektpartner HSVA ein anwendungsorientierter Testfall erarbeitet, um die mit dem Schiffsmodell mo4977 von der HSVA im Eistank durchgeführten Schleppversuche numerisch nachbilden zu können. Das Hauptinteresse seitens der HSVA lag hierbei in der Messung des Eisvolumenstroms im Zufluss zum Propeller.

Entwickelt wurde ein entsprechender numerischer Testfall in e1be, der sich aus mehreren, teils stark interferierenden Modulen zusammensetzt. Die besondere Herausforderung dabei war, die Module harmonisierend zu gestalten, um eine möglichst lange Simulationsdauer zu erzielen, ohne dass das Simulationsprogramm aufgrund von unvorhersehbaren Komplikationen (numerischen Instabilitäten) vorzeitig abbricht.

Der Testfall umfasst die Simulation des mo4977-Schiffsmodells (vgl. Abb. 39) im Modellmaßstab (1:19) in Fluid-Eis-Interaktion, die eine dynamische Zu- und Abfuhr von Eisschollen, die Berücksichtigung eines vereinfachten Propulsionsmodells zur Nachbildung zweier Pod-Antriebe sowie die Messmethode zur Ermittlung des Eisvolumenstroms und die Modellierung eines Eis-Fernfeldes umfasst.



Abbildung 39: Formgebung des mo4977-Schiffsmodells inklusive Pod-Gondeln und Propeller.

Der Testfall wurde iterativ entwickelt und schloss zu Beginn die Berücksichtigung der freien Wasseroberfläche und die Nutzung lokaler Gitterverfeinerung um die Pod-Gondeln mit ein. Da allerdings eine Eisschollenzufuhr auf Höhe der Wasserlinie dazu führte, dass kaum Eisschollen in den Bereich der Propeller gelangten und seitens des Projektpartners HSVA Eisschollen in diesem Bereich erwartet wurden, wurde in Absprache mit der HSVA die Höhe der Zufuhrposition auf die Kielline (*Baseline*) des Modells verlegt. Dies erlaubte die Reduktion des Rechengebietes auf das tief-getauchte Hinterschiff und führte zu einer insgesamt feineren Diskretisierung des Rechengebiets, vgl. Abb. 40. Im Gegenzug machte dies die Simulation der freien Wasseroberfläche, die Modellierung des Fernfeldeises als auch die Nutzung einer lokalen Gitterverfeinerung überflüssig.

Das Modell der mo4977 ist 4,61 m lang und 0,98 m breit. Der Tiefgang wurde für die zu untersuchende Konfiguration auf 0,35 m festgelegt. Für das Strömungsberechnungsgebiet wurde ein Abmaß von 3,2 m \times 1,6 m \times 0,6 m gewählt, vgl. Abb. 40. Eine Diskretisierung durch

$800 \times 400 \times 151 = 48\,320\,000$ Gitterknoten resultierte in einer Gitterweite von $\Delta x = 0,004$ m. Auf Basis einer Einlassgeschwindigkeit von $v_{\text{inlet}} = 0,5$ m/s, einer maximal auftretenden Geschwindigkeit von $v_{\text{max}} = 2$ m/s und einer Machzahl von $\text{Ma} = 0,05$ ergab sich eine Zeitschrittweite von $\Delta t = \frac{\Delta x \text{Ma}}{\sqrt{3} v_{\text{max}}} = 5,78 \cdot 10^{-5}$ s. Für das Wasser wurde eine Dichte von $\rho = 1000$ kg/m³ und eine Viskosität von $\nu = 1,000\,46 \cdot 10^{-6}$ m²/s angenommen. Unter Wahl des charakteristischen dem Strömungsproblem zugrundeliegenden Längenmaß von $l_{\text{char}} = 4,6$ m resultiert eine Reynolds-Zahl von $\text{Re} = \frac{v_{\text{max}} l_{\text{char}}}{\nu} = 9,2 \cdot 10^6$. Die Strömungsrichtung entspricht der positiven x -Richtung.

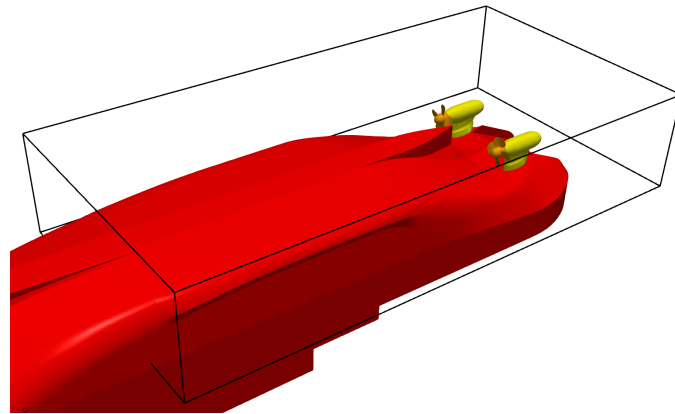


Abbildung 40: Das Strömungsberechnungsgebiet (schwarz umrandete Box) des mo4977-Modells beschränkt sich auf das tief-getauchte Hinterschiff.

Zur effizienten und möglichst problemlosen Simulation mussten vor Beginn die seitens der HSVA zur Verfügung gestellten Geometrien aufbereitet, ausgedünnt und repariert werden. Dies betraf das Schiffsmodell mo4977, die zwei Pod-Antriebe (bestehend aus Schaft, Gondel, Haube und Propellerblättern mit einem Durchmesser von $D = 0,185$ m, vgl. Abb. 41) sowie die vorgeschrittenen bzw. rautenförmigen Eisschollengeometrien.

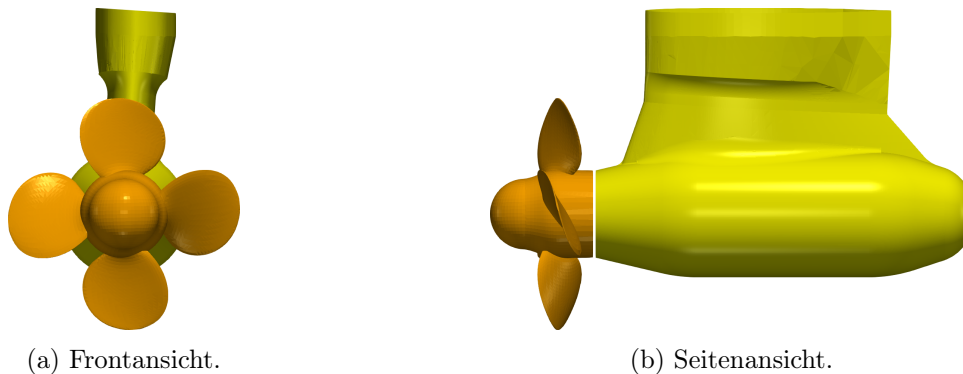


Abbildung 41: Geometrien des Pod-Antriebs auf Steuerbordseite, bestehend aus Schaft, Gondel, Haube und Propellerblättern mit $D = 0,185$ m.

Die Modelleisschollen wurden in ihrer Dicke mit $0,035$ m und $0,056$ m variiert und besitzen eine Länge von $0,36$ m sowie eine Breite von $0,19$ m, vgl. Abb. 42.

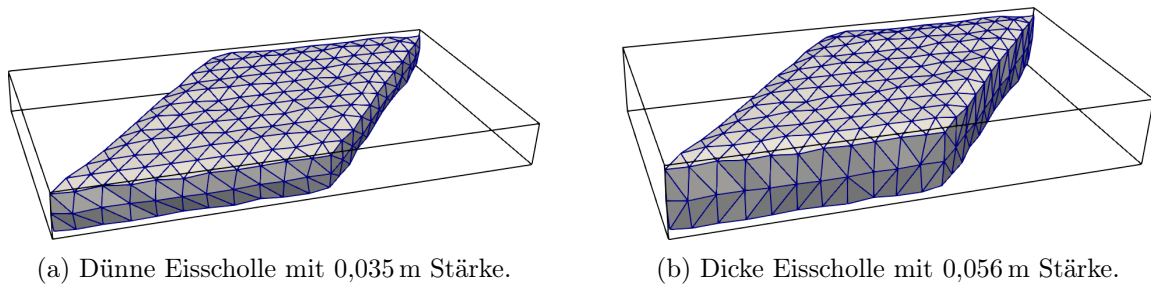


Abbildung 42: Geometrien der vorgeschnittenen (*Presawn*-Eisbedingungen) bzw. rautenförmigen Modelleisschollen.

Basieren auf einer Dichte von $\rho = 920 \text{ kg/m}^3$ wurden Masse und Trägheit der Eisschollen bestimmt. Um eine möglichst physikalisch konsistente Simulation schwimmender Mehrkörpersysteme, wie der Interaktion zwischen Schiff, drehendem Propeller, Eisschollen und Wasser, ausführen zu können kamen die Methoden der Kollisionsdetektion und -auflösung der *Open Dynamics Engine* aus Kapitel 4 zur Anwendung. Um eine beschleunigte Kollisionsdetektion zu erzielen, wurde der *SAPSpace*-Algorithmus gewählt. Als Reibungsmodell kam das bedeutend genauere Modell №1 zum Einsatz. Zwischen allen Kollisionspaaren wurde der Reibungskoeffizient $\mu = 0$ gewählt, um ein etwaiges Abbremsen und/oder Anhaften der Eisschollen an dem Schiffsrumpf zu vermeiden. Für den Restitutionskoeffizienten wurde $k = 0,001$ angenommen. Als Kraftberechnungsmethode wurde das Spannungsintegrationsverfahren über die Dreiecke der Geometrie aus Kapitel 5.1.3 gewählt.

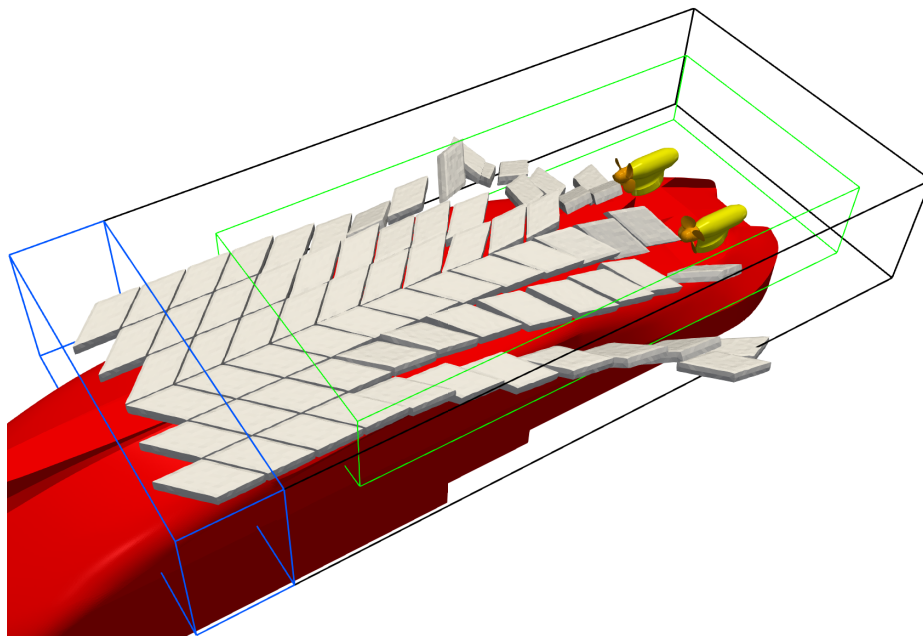
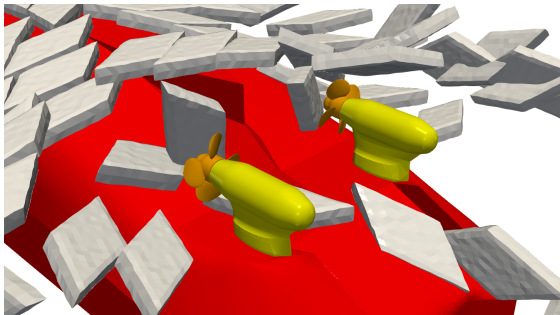


Abbildung 43: Anwendung des Eisschollengenerators: In der Initialisierungszone (blaue Box) werden Eisschollen hinzugefügt und werden kinematisch in das Strömungsberechnungsgebiet (schwarze Box) bewegt, bis sie die dynamische Zone erreichen (grüne Box) und komplett dynamisch mit anderen Eisschollen, dem Schiff inkl. Pod-Antrieb und dem umgebenden Wasser interagieren können.

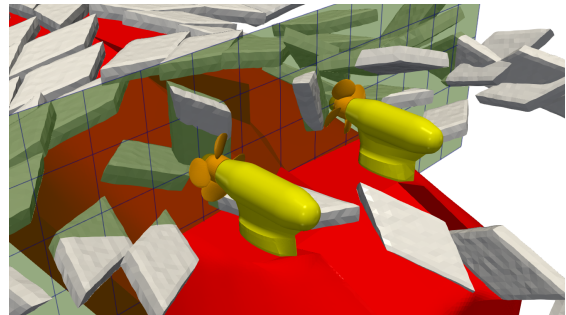
Zur kontinuierlichen Eisschollenzu- und -abfuhr wurde der in Kapitel 8 vorgestellte Eisschollengenerator verwendet. Die Initialisierungszone befand sich unterhalb der Kiellinie des Rumpfes auf etwa $\frac{1}{3}$ der Schiffslänge und schloss sich an das Strömungsberechnungsgebiet an, vgl. Abb. 43. Die Initialisierungsfrequenz der rautenförmigen Eisschollen wurde an die Einstömungsgeschwindigkeit angepasst, sodass die Eisschollen möglichst störungsfrei mit der gleichen Geschwindigkeit wie die der Strömung in das Strömungsberechnungsgebiet kinematisch geführt wurden und nur ein minimaler Abstand zwischen den einzelnen Schollen bestand. Je Schiffseite wurden zu gegebenem Zeitpunkt drei Eisschollen hinzugefügt. Sobald die Eisschollen die dynamische Zone erreichten, konnten diese komplett dynamisch mit anderen Eisschollen als auch mit dem Schiff und dem Pod-Antrieb sowie dem umgebenden Wasser interagieren. Nach dem Verlassen der dynamischen Zone wurden die Eisschollen wieder kinematisch geführt, bis sie sich außerhalb des Strömungsberechnungsgebiet befanden und zerstört werden konnten.

Zur Nachbildung des Propulsionsverhaltens der Pod-Antriebe wurde die in Kapitel 7 vorgestellte LibPPB-Bibliothek im Zweischrauber-Betrieb verwendet. Die Propellerdrehzahl n wurde in drei Stufen variiert: $n \in \{ 5,750; 9,009; 18,018 \} /s$, sodass sich drei Fortschrittzahlen $J = \frac{v_{inlet}}{nD} \in \{ 0,15; 0,30; 0,47 \}$ ergaben. Die induzierten Geschwindigkeiten wurden im Momentenraum aufgeprägt. Da die LibPPB zur Bestimmung der induzierten Geschwindigkeiten auf ein Quellfeld ohne Strömungshindernisse (mit Ausnahme der Eisschollen) angewiesen ist und die mo4977 im Bereich vor den Pod-Antrieben starke Krümmungen in der Außenhaut aufweist, sind Quell- und Induktionsebene um $\Delta z = 0,02\text{ m}$ in der Vertikalen verschoben, vgl. Abb. 31.

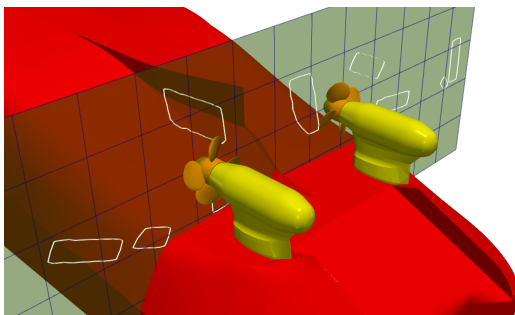
Außerdem kam die in Kapitel 9 vorgestellte Messmethode zur Bestimmung des Eis-Volumenstroms zur Anwendung. Als Messebene fungierte die gesamte y - z -Ebene $\frac{r}{4}$ stromauf vor dem Propeller. Eisschollen, die sich in der kinematischen Zone befanden, wurden von der Messung ausgeschlossen. Die Abb. 44 zeigt Schritt für Schritt die mit der Messmethode gewonnenen Ergebnisse zu einem ausgewählten Zeitschritt. Die mit dieser Messmethode gewonnenen Daten wurden für alle Variationen und Permutationen der Eisschollendicke und Propellerdrehzahlen an die HSVA übermittelt.



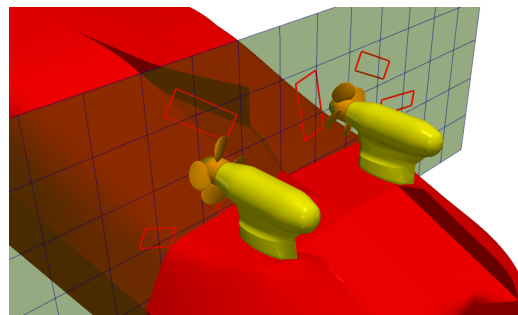
(a) Verteilung der Eisschollen zu einem ausgewählten Zeitschritt.



(b) Die Messebene (grün) befindet sich knapp vor dem Propeller.



(c) Schnittmenge (weiß) zwischen Eisschollen und Messebene als Polygonzug.



(d) Diskretisierung der Schnittmenge auf einen zweidimensional orientierten Hüllkörper mit gleicher Fläche (rot). Berücksichtigt werden nur dynamisch geführte Eisschollen.

Abbildung 44: Verfahrensschritte der Eis-Volumenstrom-Messmethode zu einem ausgewählten Zeitschritt von (a) bis (d). Dynamische Eisschollen, deren Geometrien die Messebene schneiden, werden zu Rechtecken mit gleicher Fläche wie die der Schnittmenge diskretisiert.

13.2 Voith-Schneider-Propeller Testfall

In diesem Abschnitt werden die Anpassungen und Erweiterungen des übergeordneten Eistestfalles in **elbe** erläutert, welche sich aus den Anforderungen der numerischer Simulation von eisgehenden mittels Voith-Schneider-Propeller angetriebenen Schiffen ergeben. In Abb. 45 ist der Vergleich einer Modellversuchsaufnahme sowie eines dazugehörigen Simulationenergebnisses unter äquivalenter Parameterwahl gezeigt.



Abbildung 45: Instantanaufnahme eines HSVA Modellversuchs (links) unter Verwendung einer entwickelten generischen *Zuführrumpf* Geometrie (Mitte), die darauf abzielt möglichst das gesamte unter den Rumpf geführte Eis zum Propulsor zu führen. Aufnahme des zur Modellversuchskonfiguration gehörigen **elbe** Simulationsergebnisses (rechts) dessen Charakteristik des Eisschollenflusses in guter Übereinstimmung mit dem real gefahrenen Modellversuches steht.

Nachdem in Kapitel 13.2.1 die numerische Umsetzung des Testfalles beschrieben wird, werden in den nachfolgenden Abschnitten umfangreiche Validierungen des Löser anhand verfügbarer HSVA Modellversuchsdatensätze gezeigt sowie die Güte der Einsetzbarkeit des Strömungslöser zur Simulation eisgehender Schiffe klassifiziert. Die Validierung erfolgt nacheinander in mehreren Schritten mit jeweils steigendem Komplexitätsgrad. Während in Kapitel 13.2.3 die Validierung des Löser für Simulationen unter Freifahrtbedingungen dargestellt wird, folgt in Kapitel 13.2.4 die Zusammenfassung einer Parameterstudie von Simulationen unter Eisbedingungen.

Im Rahmen der numerischen Analysen des Voith-Schneider Eis-Testfalles wurden zwei grundlegend verschiedene Rumpfformen herangezogen, vgl. Abb. 46. Einerseits wurde ein generischer Rumpf verwendet, der speziell für den HSVA Modellversuch konzipiert wurde und darauf abzielt möglichst das gesamte Eis aus dem Zustrom zum Rumpf bis hin zum Propulsor zu führen. Im Rahmen dieser Parameterstudie werden charakteristische Größen – wie bspw. die homogene Anströmgeschwindigkeit, die VSP Drehzahl als auch die Eisbedingungen – verändert und es können Trends abgeleitet werden. Andererseits werden Simulationen der realen Schiffsgometrie des Voith Wassertreckers (VWT) unter Eisbedingungen ausgeführt, deren Analyse das Kapitel 13.2.5 beinhaltet.

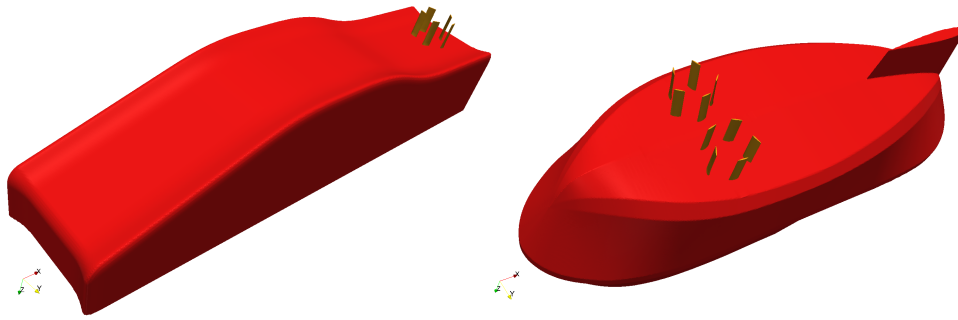


Abbildung 46: Darstellung der zwei unterschiedlichen untersuchten Rumpfgeometrien. Generische *Zuführrumpf* Geometrie: Ein VSP Radkörper im Heckbereich angeordnet (links). Reale Geometrie des Voith Wassertrecker (VWT) mit der Anordnung von zwei gegenläufig drehenden VSP Radkörpern auf ca. $1/3$ der Schiffslänge (L_{pp}) vom vorderen Lot (rechts).

13.2.1 Numerische Umsetzung des Testfalles

Zur Umsetzung des Testfalles wurde ein Bewegungsmodul in den Strömungslöser implementiert, das die reale Voith-Schneider-Propeller Schubkurbelkinematik, d.h. die Bewegung der VSP Profile gegenüber der Tangente des Radkörperdrehkreises – die sog. Flügelwinkelkurve – abbildet. Exemplarische Flügelwinkelkurven der Backbord VSP Profile aus der Simulation im Vergleich zu dem Referenzdatenverlauf von Voith (in schwarz) sind in Abb. 47 (oben rechts) dargestellt.

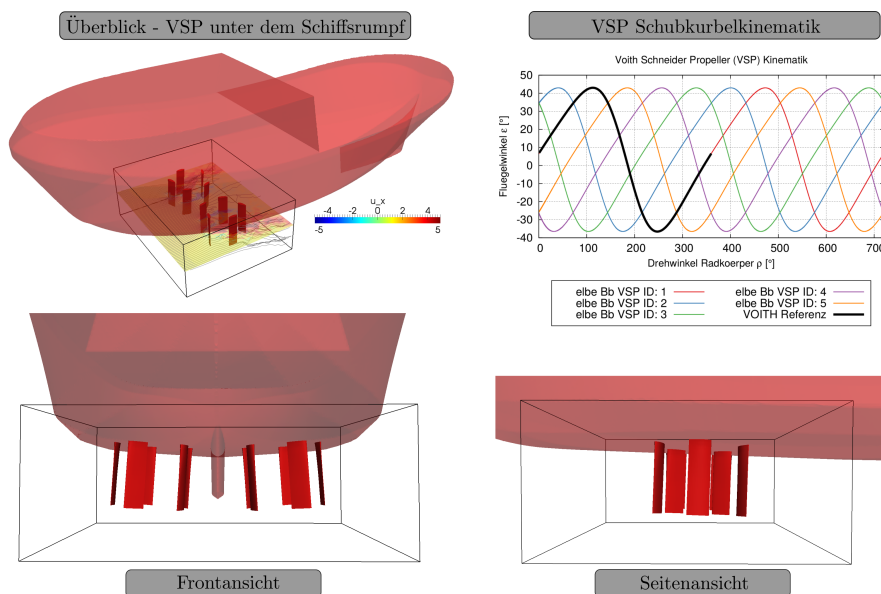


Abbildung 47: Numerische Umsetzung des Voith Wassertrecker (VWT) Testfalles: Unter dem Schiffsrumpfmodell positionierte VSP Radkörper umgeben von einer lokal feiner diskretisierten Zone – angedeutet durch den schwarz umrissenen Quader – (oben links); Validierung des implementierten Bewegungsmoduls der Schubkurbelkinematik der VSP Profile. Herausgenommen dargestellt ist der exemplarische Vergleich von Voith Referenzdaten mit der Bewegung der Backbord Profile innerhalb der Simulation (oben rechts); Front- und Seitenansicht der unter dem Rumpf angeordneten VSP Profile – zwei Radkörper mit jeweils fünf Flügeln – mit Anstellung der Drehachsen in Abhängigkeit der Außenhautnormalen des Rumpfes (untere Abbildungen).

Die Entwicklungsarbeiten umfassten hierbei zunächst einfache Tests anhand der Bewegung eines einzelnen Flügelprofils. Die in `elbe` implementierte Kinematik wurde in enger Zusammenarbeit mit dem Projektpartner Voith auf ihre Korrektheit hin überprüft. Anschließende Implementierungen stellen die Erweiterung der Kinematik auf einen sowie auf zwei VSP Radkörper (vgl. Abb. 47 (oben links) – Steuerbord und Backbord Radkörper mit jeweils fünf Profilen) sowie die Anstellung der VSP Drehachsen aufgrund der Anpassung an die Außenhautnormale des Rumpfes dar, vgl. Abb. 47 (unten).

Zur Definition des numerischen Testfalles wurden die VSP Radkörper entsprechend der Informationen aus Generalplänen positioniert, die von Voith zur Verfügung gestellt wurden. Die in `elbe` verwendete VSP Flügelgeometrie ist in Abb. 48 gezeigt. Die vom Projektpartner übermittelte triangulierte `stl` Geometrie wurde für die Simulation aufbereitet, indem die Profilhinterkante auf die in etwa in der Simulation erreichbare räumliche Diskretisierung abgeschnitten wurde, vgl. Abb. 48 (rechts). Dies zielt darauf ab, dass das sogenannte Geometriemapping – die voxelbasierte Abbildung der Geometrie in dem diskreten Rechengitter – sowie die Berechnung der Zwischengitterabstände zu keinen Problemen führt.

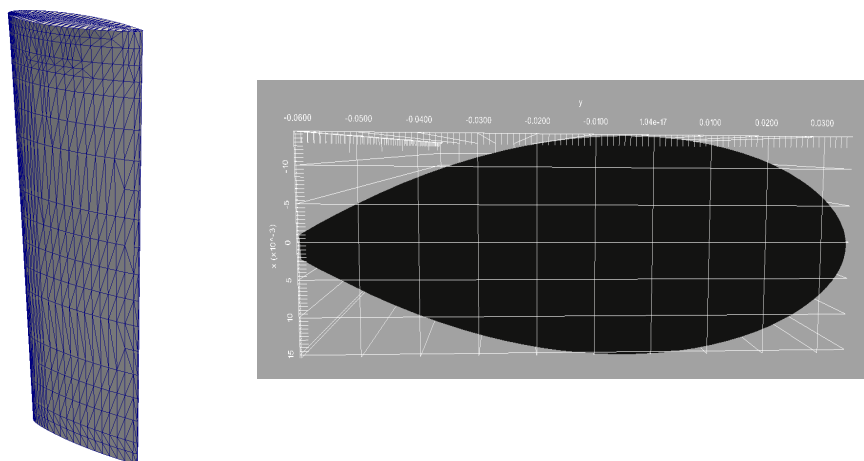


Abbildung 48: Triangulierte Geometrie des VSP Flügels aus ≈ 2400 Dreiecken bestehend (links) – Profilabmaße im Modellmaßstab ca. $0,11 \times 0,325 \times 0,04\text{m}^3$ (Sehnenlänge \times Profilhöhe \times Profildicke). Draufsicht mit abgeschnittener Profilhinterkante (rechts).

Die Abb. 49 gibt einen Überblick über die verwendeten Koordinatensysteme (KOS) sowie den Drehsinn der VSP Radkörper. Im globalen KOS zeigt die x -Achse in die negative Schiffslängsrichtung und die z -Achse ist entgegen der Erdbeschleunigung ausgerichtet. Nach dieser Definition dreht der Steuerbord (STB) Radkörper im mathematisch negativen Drehsinn um z , der Backbord (BB) Radkörper entsprechend mathematisch positiv. Die globalen Koordinaten werden bspw. für die Auswertung des global integrierten Schubes des gesamten Propulsors verwendet. Für die detaillierte Analyse in Form der Erstellung von Histogrammen lokaler Flügelkräfte und -momente – vgl. Kapitel 13.2.4 – wird ein zweites, lokales KOS verwendet. In diesem wird der lokale Profilwiderstand in Sehnenrichtung in positiver Richtung zur Profilverderkante via v

indiziert und der normal dazu ausgerichtet lokale Profilauftrieb (u) zeigt positiv aus dem VSP Radkörper Drehkreis.

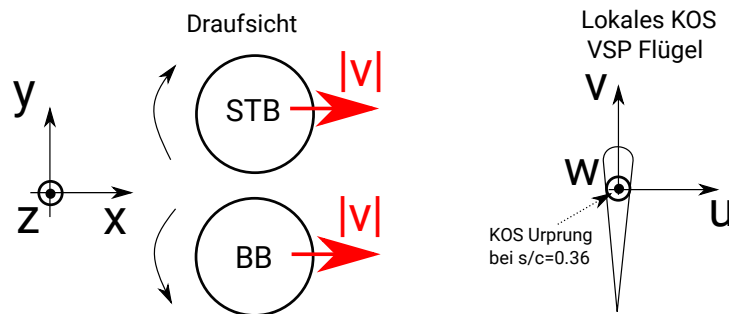


Abbildung 49: Globales Koordinatensystem (x - y - z), Drehsinn der VSP Radkörper auf Steuerbord und Backbord, Richtung der durch den Propulsor induzierten Geschwindigkeiten und lokales VSP Flügelkoordinatensystem (u : lokaler Profilauftrieb; v : lokaler Profilwiderstand).

Für den VSP Testfall werden vorgebrochene quaderförmige Eisschollen verwendet. In Abb. 50 ist die für die VSP Simulationen verwendete triangulierte generische Eisschollengeometrie dargestellt. Diese ist dem quaderförmigen Bruchmuster der HSVA VSP Modellversuche (*Presawn* Eisbedingungen) nachempfunden. Die Festlegung der Form erfolgt hierbei derart, dass der Rumpf vor dem eigentlichen Modellversuch in eine geschlossene Eisdecke (engl. *Level-Ice*) gefahren und anschließend vorsichtig zurückgezogen wird. Es folgt eine Analyse des Bruchmusters und eine Ableitung der finalen Form- und Größenangabe, der für den jeweiligen Rumpf typischen Eisscholle. Diese stellt die Grundlage der sogenannten *Presawn* Eisversuche dar, bei denen ein Schiffmodell unter eigener Propulsion durch eine vorbereitete Eisrinne definierter Eisschollen fährt.

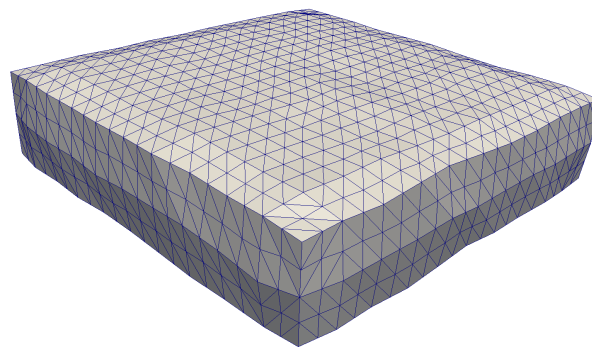


Abbildung 50: Triangulierte Eisschollengeometrie bestehend aus ≈ 2300 Dreiecken.

Dieses Verfahren zielt darauf ab den Vorgang des Eisbrechens außer Betracht zu lassen und ausschließlich das Abtauchen, das Entlanggleiten am Rumpf sowie die Propeller-Eis-Interaktion zu studieren. Für die *elbe* Simulationen sind die Modellversuche mit vorgebrochenen Eisschollen insofern vorteilhaft, da mittels *elbe* ausgenommen Starrkörper von der *Open Dynamics Engine* behandelt werden können und das Eisbrechen aus einer geschlossenen Eisdecke nicht abgebil-

det werden kann. Demnach kommen die real unter den genannten Bedingungen ausgeführten Versuche den Modellbildungen im Rahmen der numerischen Simulation am nächsten. Für alle ausgeführten VSP Simulationen finden die quaderförmigen Eisschollen Verwendung und werden für Parameterstudien in der Dicke wie auch der Kantenlänge skaliert.

Für die numerischen Simulationen befindet sich das Schiffmodell jeweils bis ca. zur Hälfte im Rechengebiet (generischer *Zuführ*-Rumpf: ab Mitte Schiff bis zum Heck; Voith Wassertrecker (VWT): Vorschiff bis Mitte Schiff) und es erfolgt keine Berücksichtigung der freien Wasseroberfläche. Das Schiffmodell ist fix und das umgebende Fluid wird unter Vorgabe einer festen homogenen Anströmgeschwindigkeit respektive eines Volumenstroms am Einströmrand des Rechengebietes bewegt.

Beim Aufsetzen des komplexen Testfalles wurden einige Behelfskörper zur Modellierung des realen Modellversuches eingeführt sowie Annahmen und Vereinfachungen zur effizienten Nutzung vorhandener Rechenressourcen (Speicherplatz wie Rechenzeit) eingeführt. Die verwendeten Behelfskörper – vgl. graue Körper in Abb. 51 – werden ausschließlich in der Kollisionsbehandlung des Starrkörperlösers *Open Dynamics Engine* berücksichtigt und werden nicht in das Fluidfeld zur Strömungslösung abgebildet.

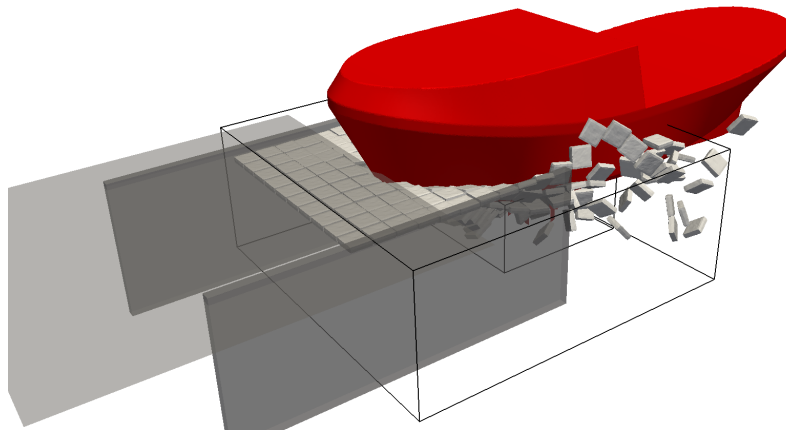


Abbildung 51: Gitter und Behelfskörper für Kollisionsbibliothek - Überblick aus der Totalen. Simulation der Voith Wassertrecker Rumpfgeometrie.

Die Körper dienen der Forcierung des Eiszustroms zum Propulsor, der Einsparung von Rechenzeit sowie der Verhinderung numerischer Instabilitäten. Zu den Körpern zählen im Detail:

- *Ebene (in x-y) über der initialen Lage der Eisschollen* (nicht reibungsbehaftet):
Erstreckung bis in etwa zum Steven des Schiffsmoделles zur Verhinderung des frühzeitigen Auftauchens der Schollen (Einphasen-Testfall, $\rho_{\text{Eis}} = 920 \text{ kg/m}^3$), Initialisierung in unmittelbarer Nähe zur Wasseroberfläche (CWL).
- *laterale Ebenen* (reibungbehaftet):
Dienen der Führung der Schollen zu den Propulsionsorganen → Minimierung der Eis-

schollenanzahl in y -Richtung bei gleichzeitiger Aufrechterhaltung der y -Erstreckung der Fluidomain, um den Einfluss des Versperrungseffektes nicht zu groß werden zu lassen (entspricht sinngemäß der Modellierung eines Eiskanals) → Einsparung von Rechenzeit.

- *Zylinder innerhalb der VSP Radkörper* (nicht reibungsbehaftet):

Verhinderung des Eindringens von Schollen in das Innere des Radkörpers, die zu einem Hin- und Herschwingen der Schollen zwischen den VSP Profilen und damit verbundenen großen Kräften / Beschleunigungen sowie dem Absturz der Simulation führen können → Die Argumentation zu dieser Modellbildung wird dadurch getragen, dass im realen Modellversuch niemals ganze Schollen in das Innere des Radkörpers eindringen können. Bei einer hohen VSP Frequenz werden die Eisschollen x -fach von den VSP Profilen getroffen werden und in (sehr) kleinen Stücken (*Milling*) advektiert. Da *Milling* für den VSP Testfall außen vor gelassen wird, ist die Verhinderung des Eindringens der Schollen in das Innere durch diese Modellbildung gerechtfertigt.

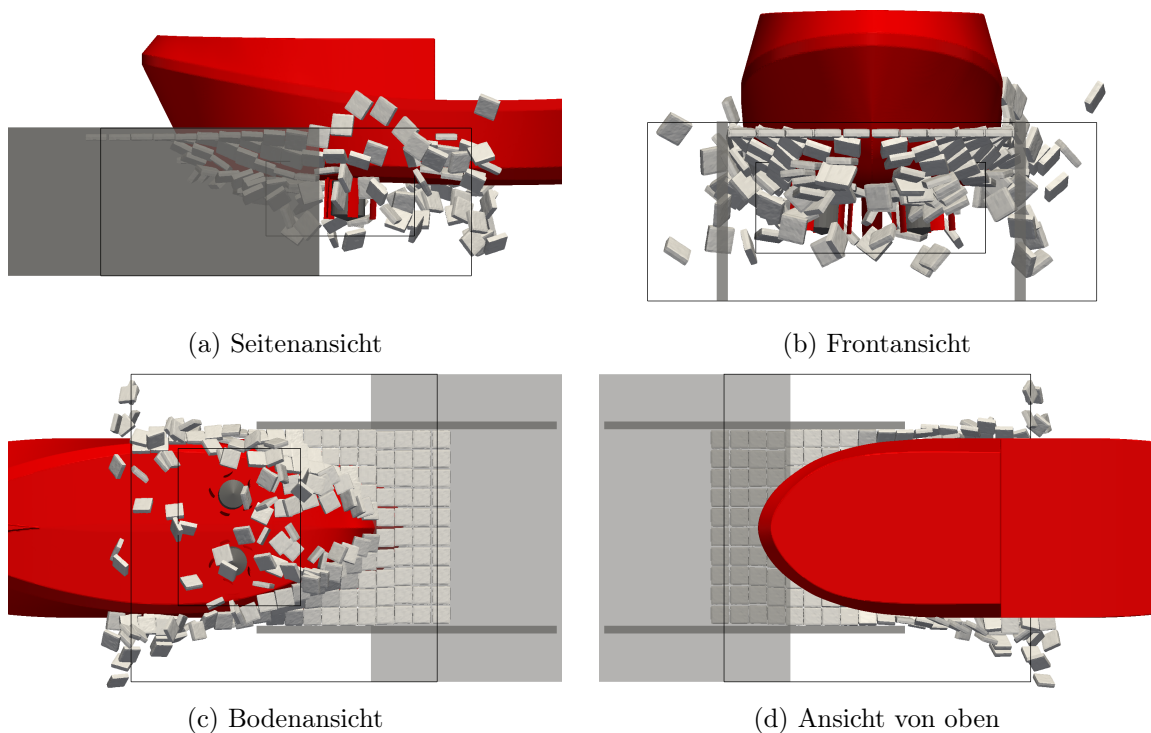


Abbildung 52: Gitter und Behelfskörper – die ausschließlich in der Kollisionsdetektion und -lösung der *Open Dynamics Engine* berücksichtigt werden und nicht als Hindernisgeometrien in das Strömungsfeld abgebildet werden – in vier verschiedenen Ansichten.

In den Abbn. 51 sowie 52 sind Eisschollen erkennbar, die sich oberhalb der Wasseroberfläche bzw. der oberen Rechengietsgrenze der Fluidomain befinden. Diese unphysikalische Position wird dadurch erklärt, dass die Eisschollen innerhalb der Simulationsumgebung kontinuierlich aus dem Rechengiet abgeführt werden müssen und erst dann gelöscht werden können, sobald ihr Schwerpunkt (oder ihr Hüllkörper) weit genug von der Rechengietsgrenze entfernt ist. In

diesem Zusammenhang werden die Schollen kinematisch – d.h. mit einer vorgeschriebenen Bewegung – aus dem Gebiet geführt, um zu verhindern, dass Gebiete aus denen Schollen zu früh gelöscht werden mit neuen Fluidknoten reinitialisiert werden müssen, wofür de facto keine validen Daten vorliegen. Die Schollenbewegung oberhalb des Rechengebietes ist demnach als keine physikalisch konsistente Bewegung anzusehen und dient lediglich der numerischen Umsetzung des Ausführens und Löschsens der Schollen aus der Simulation.

13.2.2 VSP Validierung und Parameterstudie für Simulationen eines generischen Rumpfes im Modellmaßstab

In diesem Abschnitt wird die Validierung der entwickelten Simulationsumgebung anhand der Studie eines generischen Rumpfes beschrieben. Der Rumpf wurde speziell dafür konzipiert, um möglichst alle im Zustrom zum Rumpf befindlichen vorgesägten Eisschollen zum Propulsor zu führen und somit eine detaillierte Analyse der Eis-Propulsor Interaktionen zu ermöglichen, vgl. Abb. 53.

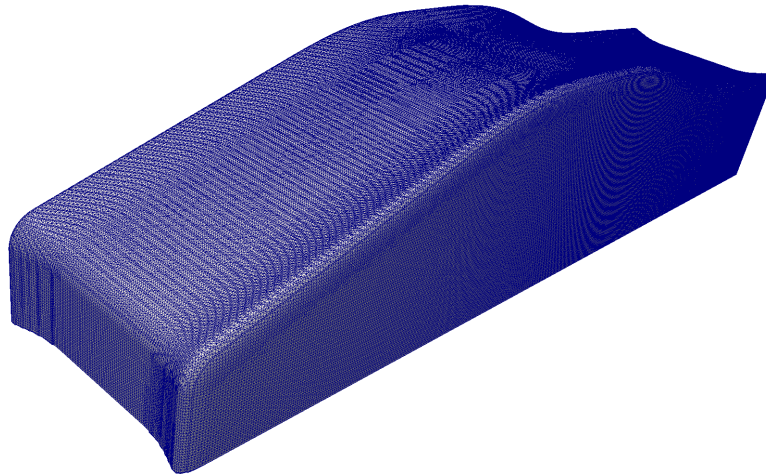


Abbildung 53: Triangulierte Geometrie des *Zuführrrumpfes* aus $\approx 350\,000$ Dreiecken bestehend. Die Hüllkörperabmessungen des Modellmaßstabes betragen $5,67 \times 1,3 \times 1,25 \text{ m}^3$ ($L \times B \times H$).

Die Propulsion wird durch eine VSP Einzel-Radkörper Anordnung im Heckbereich des Rumpfes realisiert. Das numerische Rechengebiet beginnt ab Mitte Schiff. Die Initialisierung der Schollen erfolgt innerhalb der Simulation ebenfalls mittschiffs direkt unterhalb des Rumpfes. Es liegen Referenzdaten der HSVA Modellversuchsmessungen in Form von an den instrumentierten VSP Flügeln gemessenen Kraft- und Momentengrößen vor. Für die Validierung ausschlaggebend sind vier Modellversuche für die unter Beibehaltung des Fortschrittsgrades J die homogene Anström- respektive Modellgeschwindigkeit, die VSP Drehzahl sowie die Kantenlänge der Eisschollen verändert wurden, vgl. Tab. 4 Sim.-Nr. 1–4.

Um mithilfe numerischer Simulationen Trends weiterer Parametervariationen ableiten zu können, wurde in Abstimmung mit dem Projektpartner Voith eine Testfallmatrix mit acht weiteren Konfigurationen definiert, vgl. Tab. 4.

Tabelle 4: Testfall-Matrix zur VSP Parameterstudie des generischen Zuführumpfes im Modellmaßstab. Mittels der numerischen Simulationen Sim.Nr. **1–4** werden die real im HSVA Modellversuch gefahrenen Parameterkombinationen zur Validierung des Strömungslösers nachgebildet. Anschließend werden die Drehzahl, der Fortschrittsgrad sowie die Eisdicke unabhängig voneinander variiert, um anhand der Simulationen Trends ableiten zu können.

Sim.-Nr.	n [Hz]	v_{inf} [m/s]	J [-]	h_{Eis} [m]	s_{Eis} [m]	Bemerkung
1	3.00	0.45	0.119	0.05	0.1750	gefahrener Versuch
2	2.25	0.34	0.119	0.05	0.1750	gefahrener Versuch
3	3.00	0.45	0.119	0.05	0.3500	gefahrener Versuch
4	2.25	0.34	0.119	0.05	0.3500	gefahrener Versuch
5	4.00	0.45	0.090	0.05	0.1750	erhöhte Drehzahl
6	3.50	0.45	0.102	0.05	0.1750	erhöhte Drehzahl
7	2.25	0.45	0.159	0.05	0.1750	verminderte Drehzahl
8	3.00	0.90	0.239	0.05	0.1750	erhöhter Fortschrittsgrad
9	3.00	1.35	0.358	0.05	0.1750	erhöhter Fortschrittsgrad
10	3.00	1.80	0.477	0.05	0.1750	erhöhter Fortschrittsgrad
11	3.00	0.45	0.119	0.75	0.2625	erhöhte Eisdicke / Kantenlänge
12	3.00	0.45	0.119	0.10	0.3500	erhöhte Eisdicke / Kantenlänge

Die folgende Auflistung bietet eine Zusammenfassung der wichtigsten Simulationsparameter sowie der benötigten Rechenzeiten, um einen Einblick in die Effizienz der entwickelten Simulationsumgebung zu ermöglichen. Die Parameter werden beispielhaft für die Sim.-Nr. **1** angegeben. Für die ausgeführten Simulationen ist die explizite Zeitschrittweite des Verfahrens geringfügig variabel. Dies liegt darin begründet, dass zum jeweiligen Simulationsstart einer Parameterkombination eine Abschätzung der maximal während der Simulation auftretenden Geschwindigkeit erfolgt, die einerseits von der freien Anströmgeschwindigkeit sowie andererseits von der Umfangsgeschwindigkeit der VSP Flügel abhängig ist.

- Die Rechengebietsabmaße betragen
 - im Fernfeld / Hintergrundgitter:
 - * $3,3 \text{ m} \times 1,8 \text{ m} \times 1,0 \text{ m}$
 - * Diskretisierung mit $425 \times 232 \times 129$ Knoten
 - * räumliche Schrittweite: $dx_{\text{grob}} = 7,78 \cdot 10^{-3} \text{ m}$
 - innerhalb der den gesamten VSP Radkörper umgebenden lokalen Gitterverfeinerung:
 - * $0,7 \text{ m} \times 0,7 \text{ m} \times 0,66 \text{ m}$
 - * Diskretisierung mit $182 \times 182 \times 170$ Knoten
 - * $dx_{\text{fein}} = dx_{\text{grob}}/2 = 3,89 \cdot 10^{-3} \text{ m}$
 - Die Gesamtanzahl der diskreten Knoten beträgt 18 350 480.
- Die explizite Zeitschrittweite ist $dt_{\text{grob}} = 1,06 \cdot 10^{-4} \text{ s}$ (in dem groben Hintergrundgitter des Fernfeldes, bzw. $dt_{\text{fein}} = dt_{\text{grob}}/2 = 5,3 \cdot 10^{-5} \text{ s}$).

- Die simulierte Echtzeit beträgt 20s (in Abhängigkeit von homogener Anströmgeschwindigkeit und VSP Drehzahl variiert diese zwischen 20-25s, s.o.). Diese Wahl ergibt sich durch die Forderung, dass in etwa 40 Perioden unter Eisbedingungen in die nachfolgenden Auswertungen eingehen sollten, da die relativen Häufigkeiten der ausgewerteten Größen ab dieser Anzahl nicht mehr signifikant veränderlich sind.
- Die Simulationszeit auf einer Nvidia Tesla K40m betrug ca. 19 Stunden, wobei die Performance/Effizienz über der Laufzeit der Simulation variabel ist und mit der in dem Rechengebiet ansteigenden Anzahl der Schollen bis hin zur Sättigung (maximale Anzahl an Schollen im Gebiet, die untereinander, mit dem Rumpf sowie dem Propulsor in Interaktion stehen können) sinkt. Die jeweils minimale und maximale Schollenanzahl bei Sättigung respektive maximale/minimale Performance (gemessen in engl. *Million Node Updates Per Second* (MNUPS), Millionen Knotenaktualisierungen pro Sekunde) der Simulationen der Parameterstudie – vgl. Tab. 4 – ergeben sich zu:
 - minimale Schollenanzahl im Rechengebiet: ca. **30** (Sim.-Nr. 4)
→ Performance: ≈ 95 MNUPS (8 Stunden Simulationszeit)
 - maximale Schollenanzahl im Rechengebiet: ca. **200** (Sim.-Nr. 10)
→ Performance: ≈ 45 MNUPS (20 Stunden Simulationszeit)

13.2.3 VSP Validierung unter Freifahrtbedingungen

In diesem Abschnitt erfolgt die Zusammenfassung der Validierung der implementierten VSP Kinematik unter Freifahrtbedingungen, d.h. ohne Eiszustrom in das Rechengebiet. Die detaillierte Analyse stützt sich auf einen ausgewählten Betriebspunkt ($v_{\text{inf}} = 0,45 \text{ m/s}$, $n = 3 \text{ Hz}$) und die Validierung der mittels `e1be` erzielten Simulationsergebnisse basiert auf einem Vergleich mit CFD Referenzdaten, die vom Projektpartner Voith zur Verfügung gestellt wurden und unter Verwendung eines RANS Löser (Comet) generiert wurden.

Als Geometrie wird der generische Rumpf – vgl. Abb. 53 – mit einer Einzel-Radkörperanordnung im Heckbereich verwendet und die Steigung des Propellers wird mit 90 % eingestellt. Für diese und alle weiteren Simulationen der ausgeführten Parameterstudie – vgl. Tab. 4 – wird der Phasenwinkel der Flügelwinkelkurve des VSPs in Propellerdrehrichtung auf den konstanten Wert von $15,5^\circ$ gesetzt. Diese Wahl führt dazu, dass die betrachteten Betriebspunkte nicht querkräftfrei sind, d.h. der Propulsor induziert zudem eine Querkraft F_y normal zur Schubrichtung F_x , vgl. Abb. 55 (f), zeitlicher Mittelwert der für den Radkörper integrierten Querkraft $|\bar{F}_y| \approx 80 \text{ N}$. Da die Bestimmung des mit dem querkräftfreien Zustand korrelierenden Phasenwinkels lediglich iterativ möglich ist, zudem vom Fortschrittsgrad J abhängt und somit für jeden Betriebspunkt neu ausgeführt werden muss, wird auf diese rechenzeitintensive Methodik verzichtet.

Die Abb. 54 stellt den zeitlichen Verlauf des globalen Schubes am Einzelflügel für den simulierten Betriebspunkt der Parameterkombination Sim.-Nr. 1 dar, vgl. Tab. 4. Die transiente Anfangsphase der Simulation (die erste Periode) ist aufgrund des Anlaufvorganges des VSPs von erheb-

lichen Störungen vernunden mit der Initialisierung ausgehend von dem Gleichgewichtszustand der LBM-Verteilungsfunktionen geprägt und wird von dem Signal abgeschnitten. Der Kraftverlauf konvergiert schnell und ist aufgrund der Verwendung eines gleitenden Mittelwertfilters (die Wahl eines Glättungsintervalles von 5 ms führt bei einer Zeitschrittweite von $dt = 1,065 \cdot 10^{-4}$ s auf ein Fenster von 47 diskreten Zeitschritten bzw. diskreten Werten über die geglättet wird) nicht von ersichtlichen Oszillationen geprägt, vgl. Abb. 54.

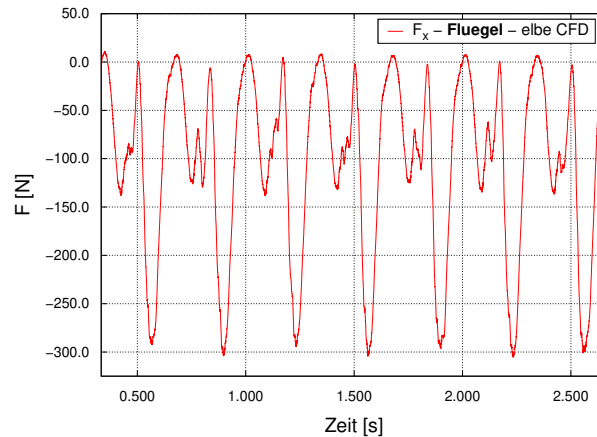
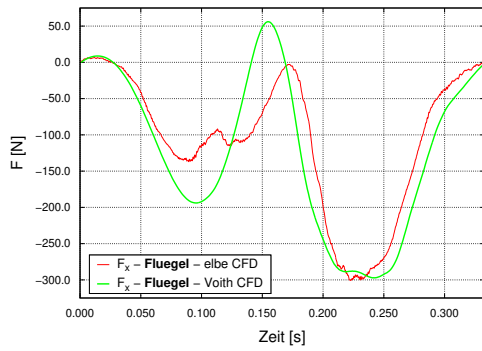


Abbildung 54: **elbe** Simulationsergebnis: Zeitverlauf der Perioden 2 – 8 ($t \in \{0.33; 2.66\}$ s) des globalen Schubanteiles F_x am Einzelflügel des VSPs für die Simulation des Betriebspunktes Sim.-Nr. 1, vgl. Tab. 4.

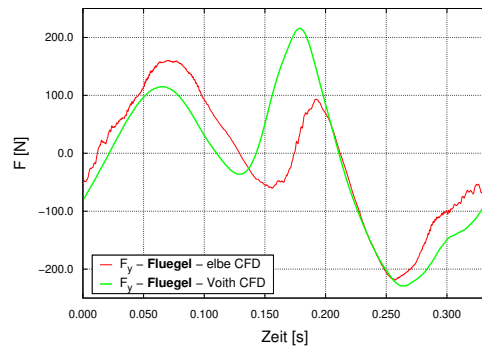
Zur detaillierten Analyse globaler wie lokaler Kräfte, Momente und der Leistungsaufnahme des VSPs wird nachfolgend die achte Periode aus Abb. 54 ($t \in \{2.33; 2.66\}$ s) separat betrachtet, vgl. Abb. 55.

Der globale Schubverlauf – Abb. 55 (a) – weist nach ca. 0,08 s (entspricht 90° Radkörperdrehwinkel) eine signifikante Abweichung zwischen **elbe** und Voith CFD auf. Während der RANS Code ab diesem Punkt einen weitere Schubzuwachs prognostiziert, bricht der mittels **elbe** berechnete Schub ein. Dies ist vermutlich auf eine Strömungsablösung aufgrund der Unterauflösung des **elbe** Rechengebietes zurückzuführen. Der lokale Profilauftrieb (c) wird mit einer guten Übereinstimmung von beiden Lösern abgebildet, während das Minimum des lokalen Profilwiderstands von **elbe** ebenfalls aufgrund des Problems der Strömungsablösung nicht getroffen wird.

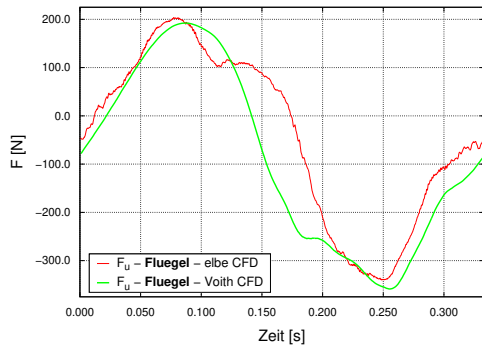
Die zeitliche Integration des Gesamt-Schubes des Radkörpers (e) führt zu einer Abweichung von ca. 18,5%. In Abb. (f) ist die oben beschriebene Thematik des nicht querkraftfreien Zustandes ersichtlich. Die Prognosen beider Löser einer mittleren Querkraft von ca. 80 N sind in guter Deckung. Die mittlere hydrodynamische Leistungsaufnahme des VSP von in etwa 1,65 kW wird ebenfalls hinreichend genau von **elbe** getroffen, vgl. Abb. 55. Aufgrund dieser initialen Analyse des VSPs unter Freifahrtbedingungen und dem Vergleich mit RANS CFD Daten wird die Schlussfolgerung getroffen, dass die VSP Hydrodynamik mit einer ausreichenden Genauigkeit von **elbe** abgebildet werden kann, sodass weitere komplexere Simulationen unter Eisbedingungen ausgeführt und untersucht werden können, vgl. Kapitel 13.2.4.



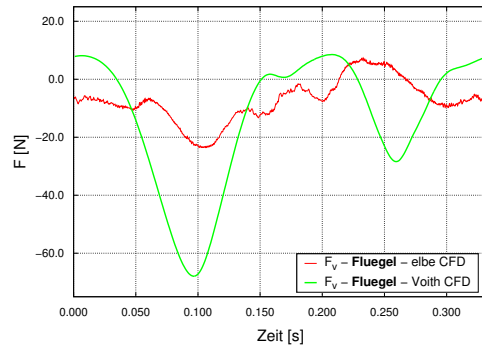
(a) Globaler Schub - Flügel



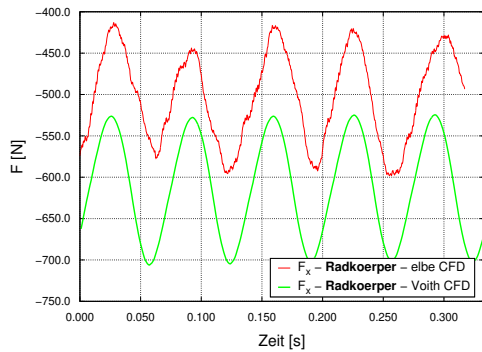
(b) Globale Querkraft - Flügel



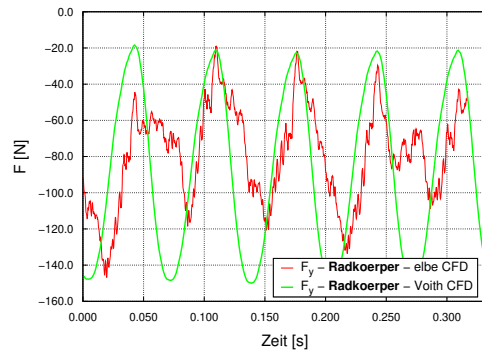
(c) Lokaler Profilauftrieb - Flügel



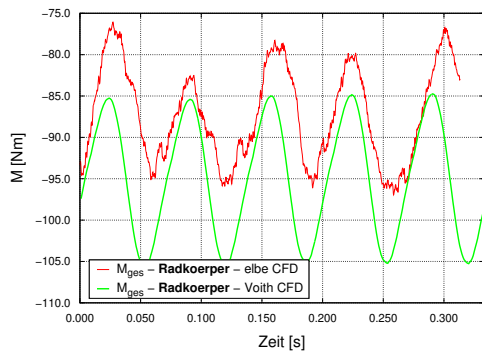
(d) Lokaler Profilwiderstand - Flügel



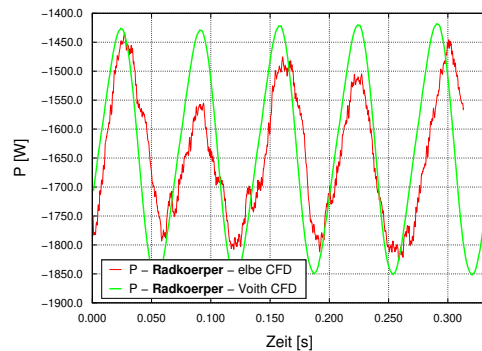
(e) Globaler Schub - Radkörper



(f) Globale Querkraft - Radkörper



(g) Gesamt-Moment - Radkörper



(h) Leistungsaufnahme - Radkörper

Abbildung 55: Detaillierte VSP Hydrodynamik Analyse (Freifahrtbedingungen ohne Eiszustrom) einer Periode für die Parameterkombination der Sim.-Nr.1 ($v_{inf} = 0,45 \text{ m/s}$, $n = 3 \text{ Hz}$) und eine Steigung des VSPs von 90 % (entspricht einer Exzentrizität von $e = 0,027 \text{ m}$). Die Plots zeigen den Vergleich der mittels **elbe** (instationäre LBM) berechneten lokalen und globalen Größen und den von Voith zur Verfügung gestellten CFD Ergebnissen (RANS-Löser). Grundlage der Auswertung ist die achte Periode nach Simulationsstart. Nach dem transienten Beginn der Simulation entspricht diese dem konvergierten Fall, vgl. Abb. 54 $t \in \{2.33; 2.66\} \text{ s}$.

13.2.4 VSP Validierung unter Eisbedingungen

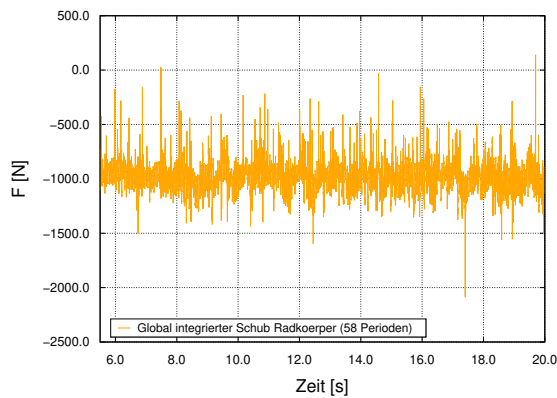
Nach der erfolgreichen Validierung des VSP Testfalles unter Freifahrtbedingungen erfolgt in diesem Abschnitt die Beschreibung einer ausführlichen Parameterstudie von Simulationen in Eisbedingungen unter Verwendung des generischen *Zuführrrumpfes* im Modellmaßstab, vgl. Abb. 53. Die detaillierte Ergebnisdarstellung der ausgewerteten globalen Größen (Schub und gesamtes Drehmoment des Radkörpers (lokales Moment um die Schaftachse des Flügels superponiert mit dem Anteil um die Rotorachse des VSP Radkörpers)) wird exemplarisch für Sim.-Nr. **5** (erhöhte VSP Drehzahl mit $v_{\text{inf}} = 0,45 \text{ m/s}$, $n = 4,0 \text{ Hz}$, $J = 0,09$, $h_{\text{Eis}} = 0,05 \text{ m}$ sowie $s_{\text{Eis}} = 0,175 \text{ m}$) gezeigt, vgl. Tab. 4.

Für die Auswertungen wird die transiente Phase, respektive die Anfangsphase der Simulation von $t = 0 \text{ s}$ bis zu dem Zeitpunkt an dem die ersten Kontakte zwischen VSP Flügeln und Eisschollen auftreten, herausgenommen. Die Dauer ist von der homogenen Anströmgeschwindigkeit abhängig und variiert für die Simulationen der Parameterstudie. Für Sim.-Nr. **5** mit $v_{\text{inf}} = 0,45 \text{ m/s}$ ergibt sich die transiente Zeitdauer unter Verwendung des Abstandes vom Einlass des Rechengebietes zum VSP Drehkreis (von ca. $2,5 \text{ m}$) zu $2,5 \text{ m}/v_{\text{inf}} \approx 5,5 \text{ s}$ (und dem anschließenden Aufrunden auf die nächste volle Periode). Auf diese Weise werden die Histogramme der Amplitudenschwankungen ausschließlich auf Basis des zusammengesetzten Signals unter Eisbedingungen erstellt. In Abbildung Abb. 56 sind links jeweils die Zeitverläufe für den, aus der Hydrodynamik und den überlagerten Kontaktkräften, zusammengesetzten Schub (gelb) sowie das Gesamt-Drehmoment dargestellt. Das Postprocessing der zusammengesetzten Signale umfasst dabei die Glättung der Hydrodynamik-Rohdaten mittels eines gleitenden Mittelwert Filters (vgl. Kapitel 13.2.3) sowie einer Normierung der aus der *Open Dynamics Engine* stammenden Kontaktkräfte/-momente. Die Normierung basiert auf einer empirisch bestimmten Vorschrift ($\Delta F_{\text{Eis}} = 1/3 c_L/u_r$, mit der Sehnenlänge c_L des Profils sowie der Umfangsgeschwindigkeit der VSP Flügel auf dem Drehkreis von $u_r = \pi n D$ mit dem Radkörperdurchmesser D) und ist als eine *gemittelte* Kontaktzeit zwischen der Eisscholle und dem VSP Flügel zu interpretieren. Hierdurch wird ein Verwischen der instantan sehr ausgeprägten Kraftspitzen am VSP erzielt, die dadurch begründet sind, dass für die ausgeführten Simulationen kein Zerteilen der Starrkörper berücksichtigt wird und es aufgrund der hohen Umfangsgeschwindigkeit des VSPs trotz der kleinen expliziten Zeitschrittweite zu großen Überschneidungen und damit hohen resultierenden Kontaktkräften kommt.

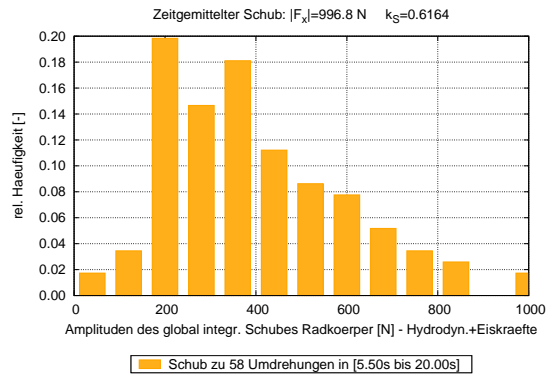
Rechts sind die aus den Zeitverläufen abgeleiteten Histogramme gezeigt. Diese werden aus den postprozessierten Verläufen erstellt, indem je Periode zwei Amplituden via Maximum-Mittelwert und Mittelwert-Minimum der jeweils betreffenden Größe ausgewertet werden.

Das Histogramm der hydrodynamischen Leistungsschwankung des VSPs ist in Abb. 57 gegeben. Für die Auswertung des **e1be** Simulationsergebnisses wird zwischen dem Moment sowie der Leistungsaufnahme der einfache Zusammenhang $P = M_{\text{ges}} 2\pi n$ angesetzt. Dieser weicht von der von Voith verwendeten Vorgehensweise insofern ab, als dass diese keine direkte Berechnung der Leistung über das einfach superponierte Gesamt-Moment vorsieht, da hierdurch die Verstellleistung

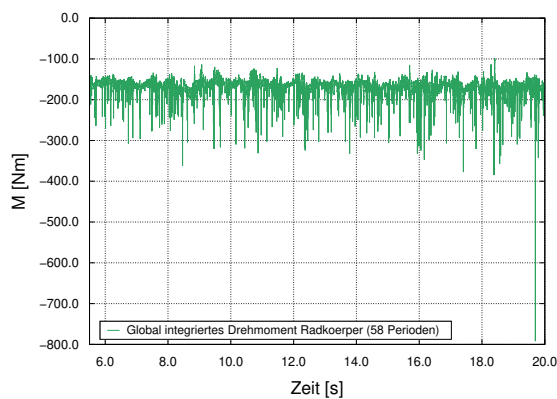
der Flügel nicht vollständig korrekt erfasst werden kann.



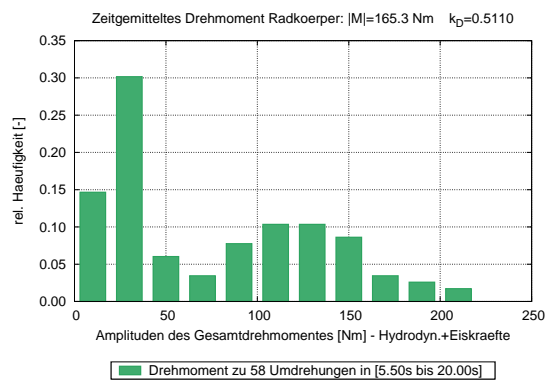
(a) Globaler Schub (Hydro+Eis)



(b) Histogramm Schubschwankungen



(c) Globales Gesamt-Drehmoment (Hydro+Eis)



(d) Histogramm Drehmomentschwankungen

Abbildung 56: links: Aus der Hydrodynamik und den Kontaktkräften superponierte Verläufe des globalen Schubes und den globalen zusammengesetzten Drehmomentes (je VSP Flügel lokales Schaftmoment sowie Anteil um die Rotorachse); rechts: zugehörige Amplitudenhistogramme zur Indikation der Schub- sowie Drehzahlschwankungen.

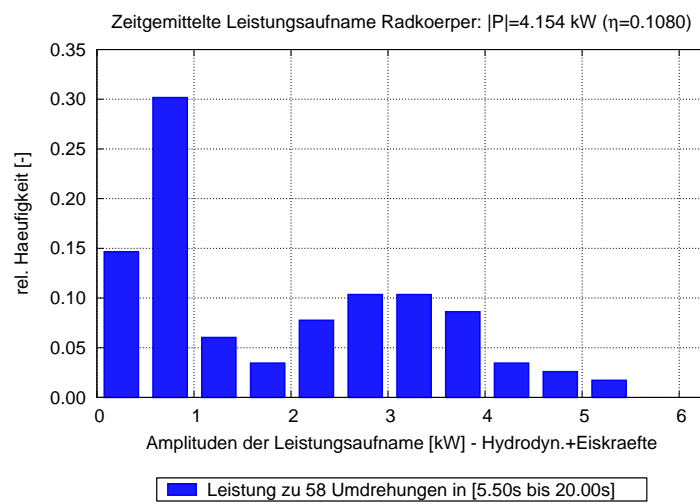


Abbildung 57: Kollektiv der Schwankungen der VSP-Radkörper Leistungsaufnahme.

Die hier vorliegende Auswertung unter Verwendung des einfachen Zusammenhanges von M_{ges} und P muss daher als Anhaltswert zur Beurteilung der Leistungsschwankungen angesehen werden.

Die Abb. 58 zeigt im Überblick eine Gegenüberstellung von Instantanaufnahmen zur Visualisierung der Scholldynamik sowie zur Darstellung der Anzahl der – bei Sättigung – im Rechengebiet befindlichen Eisschollen für alle 12 Simulationen der ausgeführten Parameterstudie, vgl. Tab. 4.

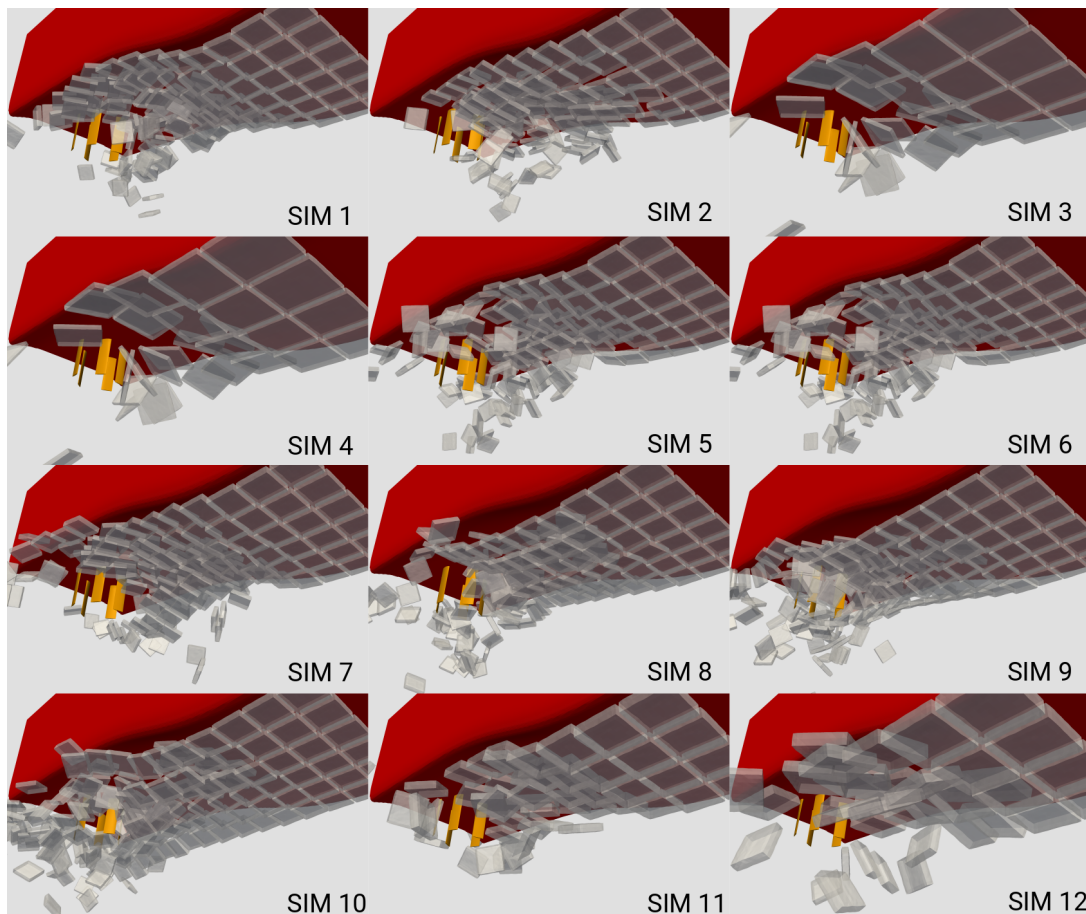
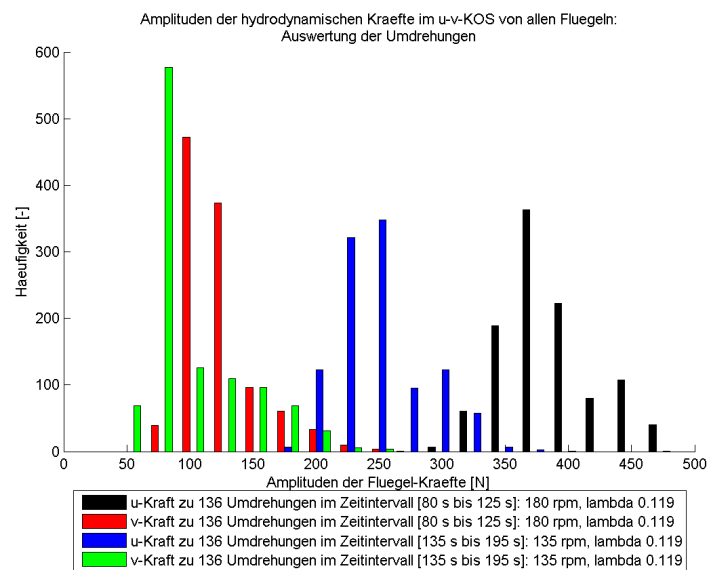


Abbildung 58: Instantanaufnahmen zur Visualisierung der Scholldynamik sowie zur Abbildung der Anzahl der im Rechengebiet befindlichen Eisschollen bei Sättigung für die Simulationen **1–12** der Parameterstudie im Modellmaßstab des generischen *Zuführumpfes*.

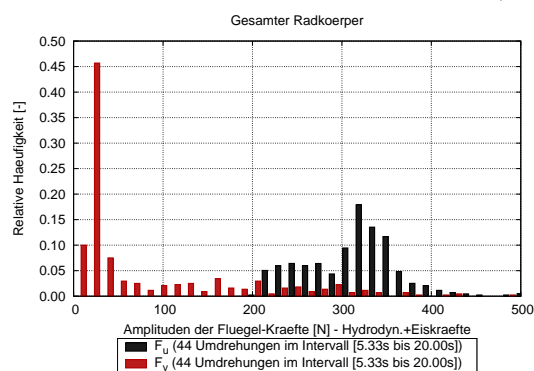
Aus Abb. 58 ist die unterschiedliche Schollenanzahl für variierende Schollengrößen sowie veränderliche Testfallparameter eindeutig erkennbar. Des Weiteren ist festzustellen, dass die Position an der die Schollen die VSP Flügel treffen zwischen den Simulationen unterschiedlich ist. Hierbei liegt der Kontaktpunkt umso tiefer (respektive umso weiter vom Flügelschaft entfernt), desto höher die homogene Anströmgeschwindigkeit gewählt wird. Dies liegt darin begründet, dass die mit einer hohen Geschwindigkeit advektierten Eisschollen in dem Abstand vom Einlass bis zum Propulsor weniger durch der Einfluss des hydrostatischen Auftriebes auftauchen und somit nicht

am Rumpf entlanggleiten (Sim.-Nr. **10**) wie dieses für geringere Anströmgeschwindigkeiten der Fall ist (Sim.-Nr. **1**).

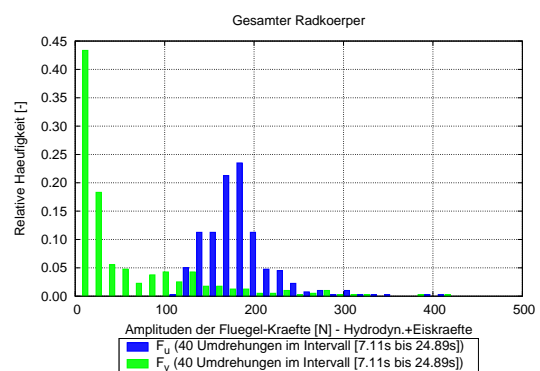
Im Folgenden wird die Validierung des LBM-Lösers anhand der aus den HSVA Modellversuchen verfügbaren Datensätze der Sim.-Nr. **1–4** (vgl. Tab. 4) beschrieben. Die Modellversuche decken zwei unterschiedliche Anströmgeschwindigkeiten des Modells sowie zwei VSP Frequenzen ab, wobei der Fortschrittsgrad mit $J = 0.119$ für alle vier Versuche konstant gehalten wird. Für die Versuche **1+2** werden Schollen der Kantenlänge $s_{\text{Eis}} = 0,175$ m verwendet, für Versuche **3+4** beträgt $s_{\text{Eis}} = 0,35$ m. In Abb. 59 (oben) sind die vom Projektpartner Voith ausgewerteten Rohdatensätze der HSVA Modellversuche in Form von Amplitudenkollektiven dargestellt.



(a) Ausgewertete Histogramme (absolute Häufigkeit) der Amplituden lokaler Flügelkräfte aus dem HSVA Modellversuch (21020 - Kantenlänge der Schollen $s_{\text{Eis}} = 0,175$ m).



(b) **elbe** Simulationsergebnis der Sim.-Nr. **1** ($v_{\text{inf}} = 0,45$ m/s, $n = 3,0$ Hz [$J = 0.119$] und $s_{\text{Eis}} = 0,175$ m).



(c) **elbe** Simulationsergebnis der Sim.-Nr. **2** ($v_{\text{inf}} = 0,34$ m/s, $n = 2,25$ Hz [$J = 0.119$] und $s_{\text{Eis}} = 0,175$ m).

Abbildung 59: Vergleich der postprozessierten HSVA Modellversuchsmessungen des instrumentierten VSPs (oben) zu den äquivalenten Parameterkombinationen der numerischen **elbe** Simulationen **1+2** (unten).

Die Histogramme in schwarz und rot entsprechen dem Versuch **1**, während die in blau und grün dargestellten Balken dem Versuch **2** zuzuordnen sind. Die Veränderung der VSP Drehzahl sowie Anströmgeschwindigkeit führt in dem Amplitudenkollektiv des lokalen Profilauftriebs (F_u) zu einer Verlagerung des Kollektivmaximums um den Faktor von ca. 1.5 (375 N/250 N). Im Vergleich zu den gleichermaßen colorierten Histogrammbalken der ausgewerteten **e1be** Simulationsergebnisse der Sim.-Nr. **1+2** wird dieser Faktor bestätigt (305 N/195 N), vgl. Abb. 59 (b)+(c).

Der Vergleich der Maxima der Kollektive zwischen den ausgewerteten Modellversuchsdatensätzen sowie den postprozessierten Simulationsergebnissen ergibt eine Abweichung der Absolutwerte der Maxima. Bspw. liegt das Maximum der u -Kraft (blau) in (a) bei ca. 250 N. In (c) wird dieses bei ca. 175 N erreicht, womit eine Abweichung von 75 N einhergeht. Diese Abweichung von in etwa 75 N liegt für alle vier in den Abbn. 59 sowie 61 vor. Eine mögliche Begründung liefert die Tatsache, dass keinerlei Filter auf das Rohdatensignal aus den Modellversuchen angewendet wurden. In Abbildung Abb. 60 sind die Graphen des Voith RANS CFD (Comet) Simulationsergebnisses und eines beispielhaften Rohdatensignals einer Periode des Modellversuches **1** dargestellt. Hierbei ist eindeutig erkennbar, dass der Rohdatenverlauf von starken Oszillationen geprägt ist. Bei der oben beschriebenen Vorgehensweise der Histogrammerstellung (je Periode Maximum-Mittelwert und Mittelwert-Minimum auswerten) werden die resultierenden Histogramme – vgl. Abb. 59 (a) sowie Abb. 61 (a) hauptsächlich von den Oszillationen bzw. Ausreißern bestimmt, die im Falle der ausgewerteten **e1be** Simulationsergebnisse unter Verwendung eines gleitenden Mittelwertfilters zur Glättung nicht vorliegen.

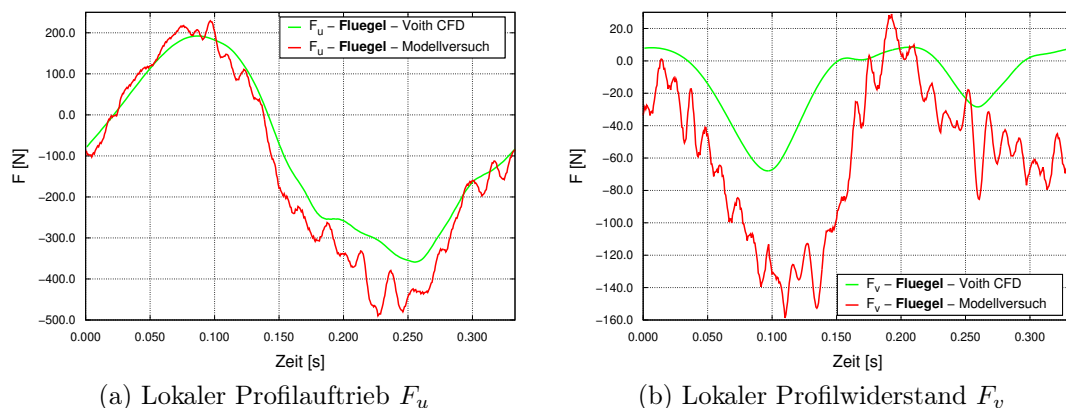
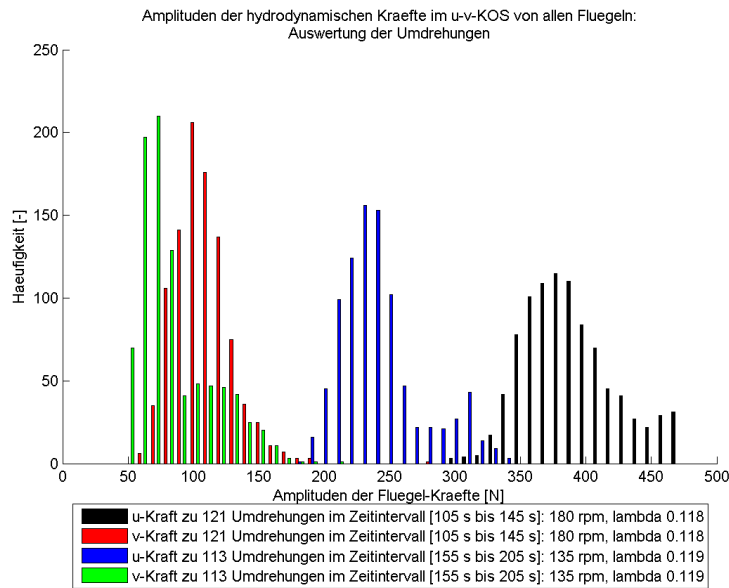
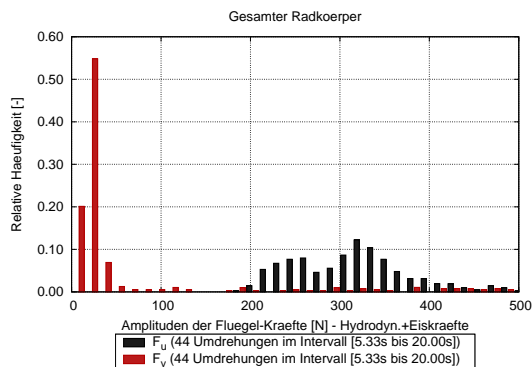


Abbildung 60: Vergleich der lokalen Flügelkräfte (lokaler Profilauftrieb F_u und lokaler Profilwiderstand F_v) zwischen den Voith Comet RANS CFD Daten sowie einer ausgewählten Periode aus dem Rohdatensignal der HSVA Modellversuchsmessungen unter Freifahrtbedingungen.

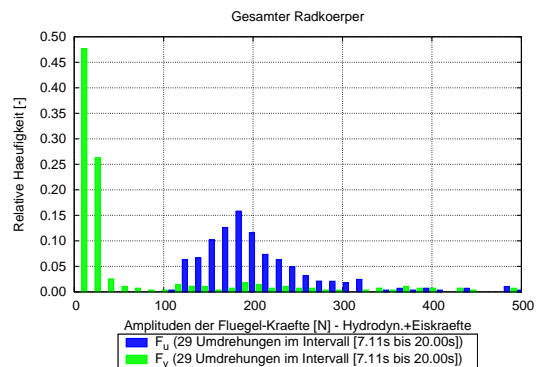
In Abb. 61 ist die äquivalente Auswertung für die Sim.-Nr. **3+4** dargestellt. Die zuvor herausgestellten absoluten Abweichungen der Maxima zwischen den Modellversuchs- und Simulationsergebnissen liegen auch für diese simulierten Betriebspunkte variabler Eisbedingungen vor.



(a) Ausgewertete Histogramme (absolute Häufigkeit) der Amplituden lokaler Flügelkräfte aus dem HSVA Modellversuch (22020 - Kantenlänge der Schollen $s_{Eis} = 0,35$ m).



(b) **e1be** Simulationsergebnis der Sim.-Nr. **3** ($v_{inf} = 0,45$ m/s, $n = 3,0$ Hz [$J = 0.119$] und $s_{Eis} = 0,35$ m).



(c) **e1be** Simulationsergebnis der Sim.-Nr. **4** ($v_{inf} = 0,34$ m/s, $n = 2,25$ Hz [$J = 0.119$] und $s_{Eis} = 0,35$ m).

Abbildung 61: Vergleich der postprocessierten HSVA Modellversuchsmessungen des instrumentierten VSP (oben) zu den äquivalenten Parameterkombinationen der numerischen **e1be** Simulationen **3+4** (unten).

Zusammenfassend wird festgestellt, dass durchaus signifikante Abweichungen zwischen den Auswertungen der Modellversuchs- und Simulationsergebnisse vorliegen. Da diese jedoch prinzipbedingt durch die Auswertungsmethoden sind, wird festgehalten, dass es möglich ist die Trends der Modellversuche (Verschiebung der Kollektiv-Maxima bei Variation des Betriebspunktes) mittels der numerischen Simulationen abzubilden. Diese Tatsache ist ausreichend, um eine Parameterstudie (Sim.-Nr. **5-12**) mittels **e1be** auszuführen.

In den folgenden Abbn. 62 und 63 sind die ausgewerteten Histogramme der lokalen Flügelkräfte F_u (Profilauftrieb) und F_v (Profilwiderstand) des gesamten Radkörpers mit fünf VSP Flügeln

dargestellt. Die vier Histogramme aus Abb. 62 (a) bis (d) entsprechen den bereits oben im Vergleich zu den Modellversuchsergebnissen ausgewerteten Histogrammen.

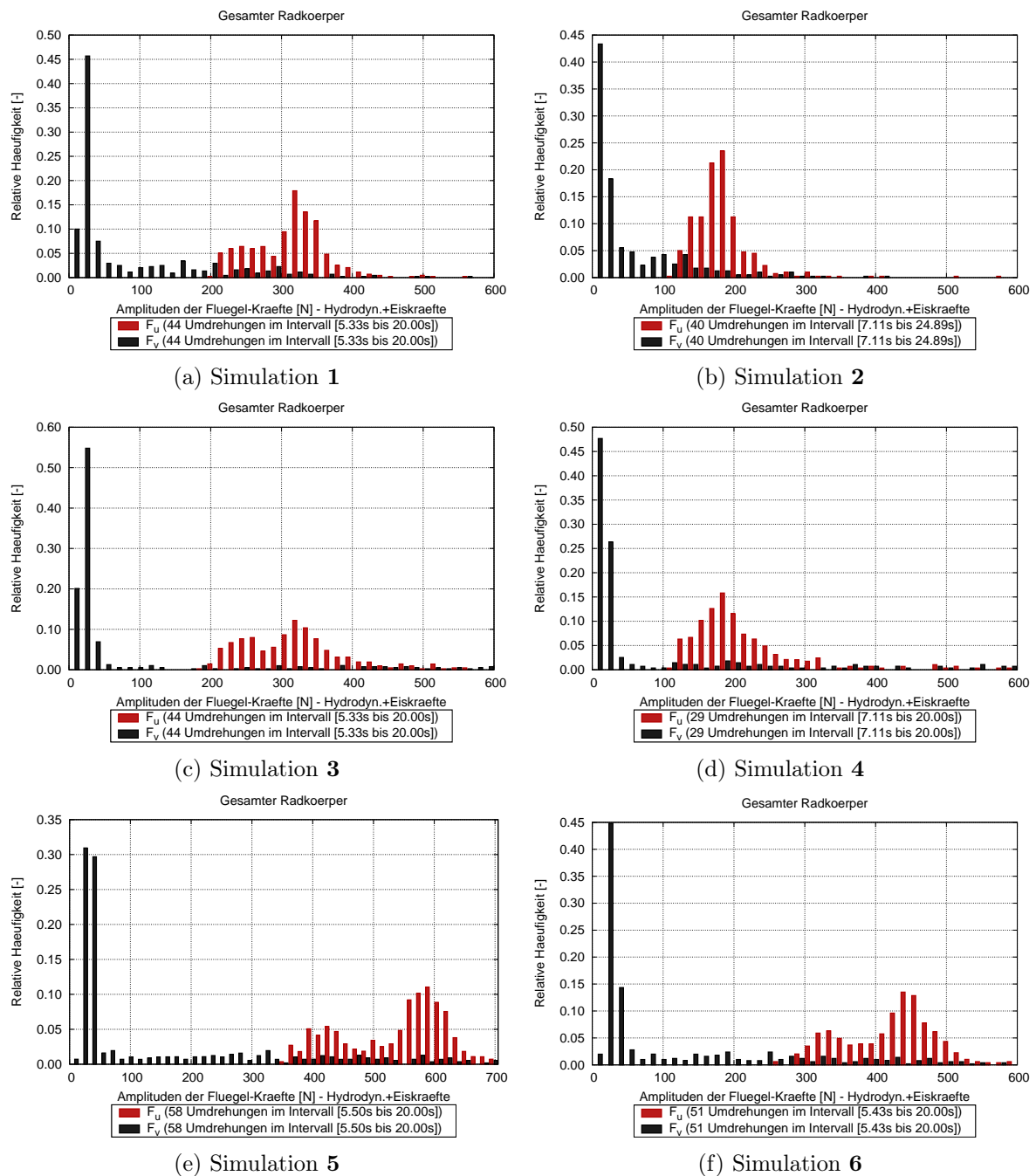


Abbildung 62: Histogramme der aus Hydrodynamik und Kontaktkräften zusammengesetzten lokalen Flügelkräfte F_u (Profilauftrieb) und F_v (Profilwiderstand) des gesamten Radkörpers mit fünf VSP Flügeln (Simulationen 1–6).

Aus den Histogrammen Abb. 62 (e) bis (f) (Sim.-Nr. 5+6) ist abzuleiten, dass die Veränderung der VSP Drehzahl den deutlichsten Einfluss auf die Lage der Kollektive besitzt. Unter Erhöhung der Drehzahl verschieben sich die Maxima der Amplituden-Kollektive deutlich nach rechts auf

der Abszisse. Für die höchste verwendete Drehzahl von $n = 4\text{ Hz}$ (e) liegt das Maximum des F_u Kollektivs bei ca. 600 N.

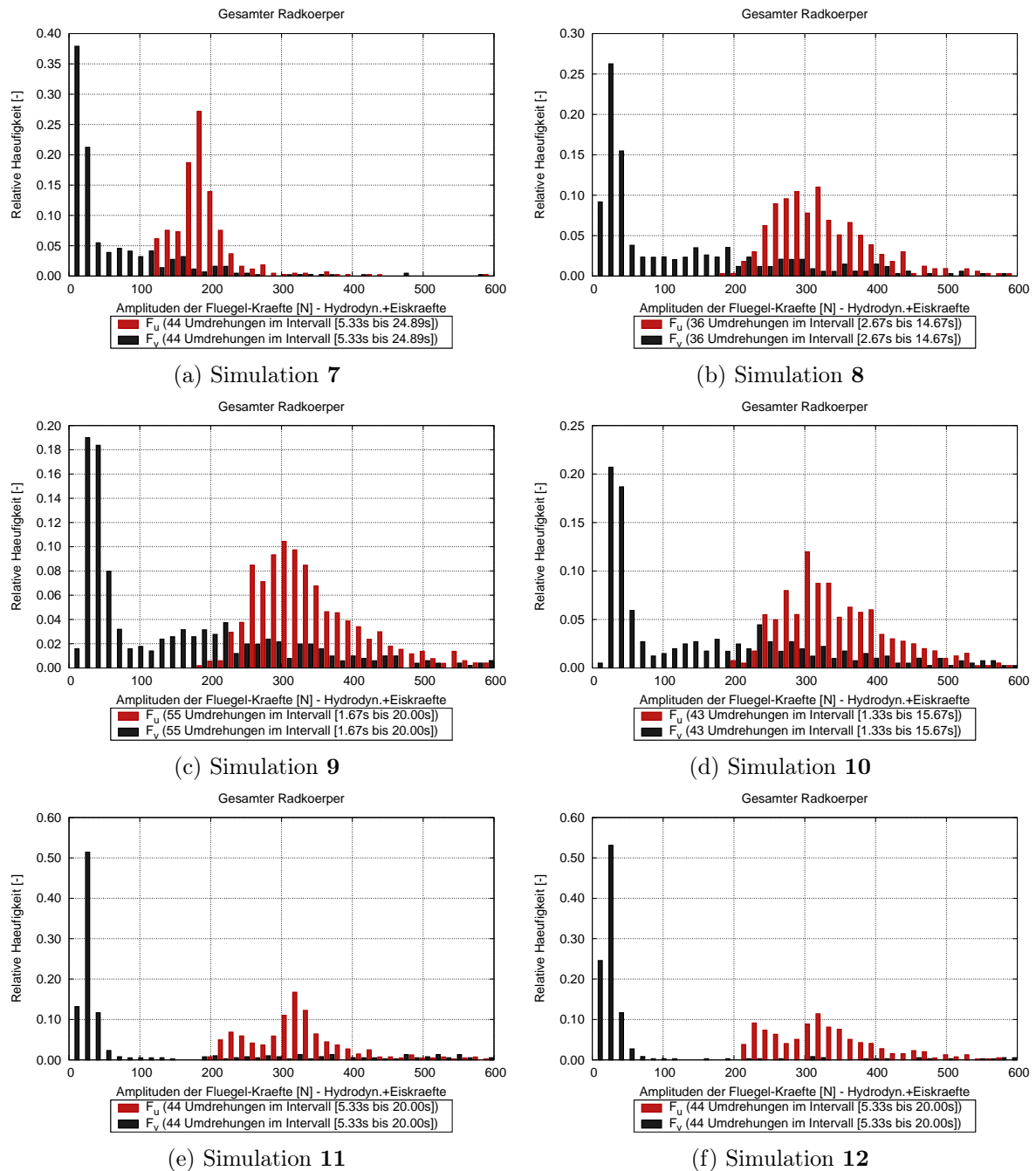


Abbildung 63: Histogramme der aus Hydrodynamik und Kontaktkräften zusammengesetzten lokalen Flügelkräfte F_u (Profilauftrieb) und F_v (Profilwiderstand) des gesamten Radkörpers mit fünf VSP Flügeln (Simulationen 7–12).

Zusätzlich zu der Auswertung der lokalen Flügelkräfte ist nachfolgend eine Zusammenfassung der Auswertung der Momente am VSP gegeben. Hierbei werden zwei Momente betrachtet – das lokale Schaftmoment des Flügels M_w sowie der Anteil um die Rotorachse des VSP Radkörpers

M_F . Das letztgenannte Moment ist dabei, aufgrund des längeren Hebels ($D/2$), deutlich größer als das lokale Schaftmoment.

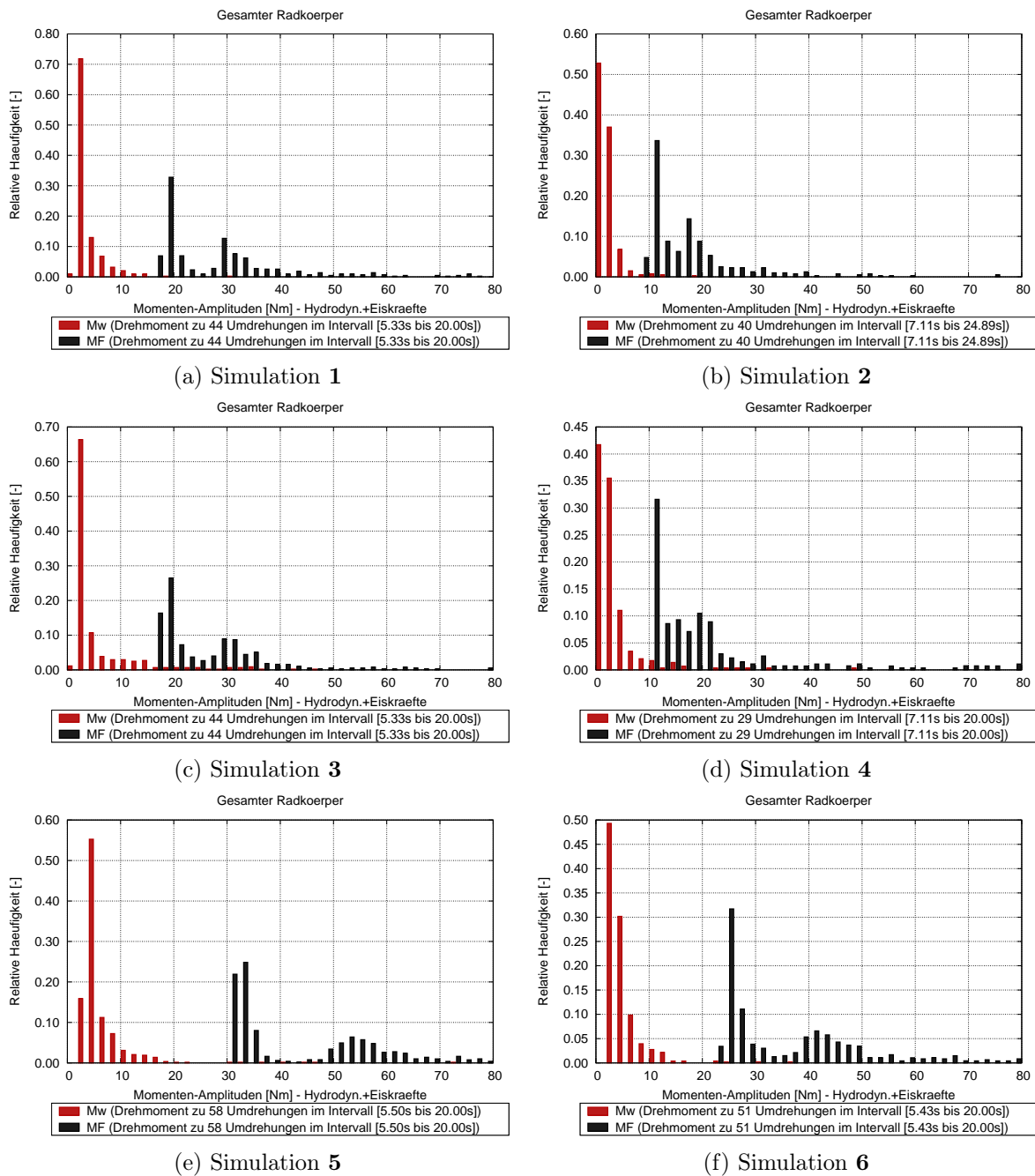


Abbildung 64: Histogramme der aus Hydrodynamik und Kontaktkräften zusammengesetzten Momente um die lokale Schaftachse (M_w) sowie dem Anteil um die Rotorachse (M_F) des gesamten Radkörpers mit fünf VSP Flügeln (Simulationen 1–6).

Aus den Abbildungen Abb. 63 (e)+(f) (Sim.-Nr. 5+6) ist eine besonders ausgeprägte Zweiteilung der Histogrammkollektive des M_F Moments mit zwei Maxima erkennbar, wobei das erste lokale Maximum bei ca. 30 N (± 5 N) eine höhere relative Häufigkeit (ca. 30%) aufweist. Die Zweiteilung liegt darin begründet, dass die Kontakte zwischen Schollen und den Flügeln immer mit

der Vorderkante erfolgen. Aufgrund der Defintion des KOS führen die Kontaktkräfte für den gegen den Uhrzeigersinn drehenden VSP stets zu negativen Beiträgen / Amplifikationen des Momentes, vgl. ebenfalls Abb. 56 (c).

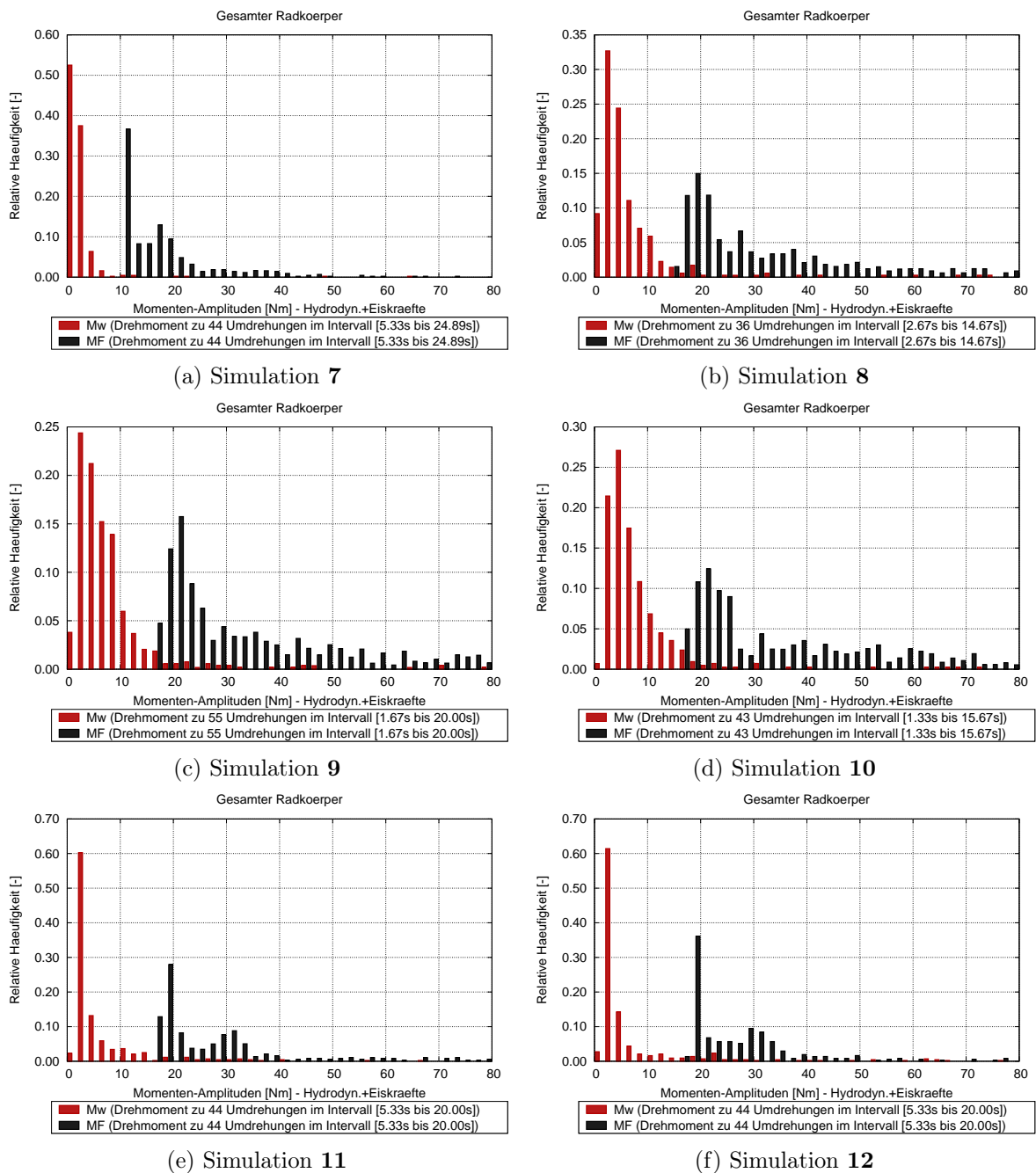


Abbildung 65: Histogramme der aus Hydrodynamik und Kontaktkräften zusammengesetzten Momente um die lokale Schaftachse (M_w) sowie dem Anteil um die Rotorachse (M_F) des gesamten Radkörpers mit fünf VSP Flügeln (Simulationen 7–12).

Die Amplituden geringeren Absolutwertes sind demnach der Hydrodynamik zuzuordnen und entsprechen den Maxima der jeweiligen Periode. Die um das zweite lokale Maximum gleichver-

teilten Amplituden höheren Absolutwertes sind Eisstößen zuzuordnen. Dass diese Ausprägung speziell bei hohen Umfangsgeschwindigkeiten auftritt, bestätigt die physikalisch konsistente Modellierung der Eis-VSP-Interaktion mittels *elbe*.

Zusammenfassend lassen sich die im Rahmen dieses Berichts dargestellten Simulationsergebnisse der für den Modellmaßstab ausgeführten Parameterstudie derart bewerten, dass es mittels *elbe* möglich ist Trends aus Variationen des VSP Betriebspunktes sowie veränderlicher Eisbedingungen abzuleiten. Weitere Detailanalysen und umfangreichere Auswertungen, die in enger Abstimmung mit dem Projektpartner Voith realisiert worden sind, gehen über den Umfang des vorliegenden Berichts hinaus.

13.2.5 Simulation der realen Voith Wassertrecker (VWT) Rumpfgeometrie

Als weiterer Schritt nach der in Kapitel 13.2.2 umfangreich beschriebenen Validierung der Eis-Simulationen des von einem VSP Radkörper betriebenen generischen *Zuführrrumpfes* erfolgt in diesem Abschnitt die Zusammenfassung der Simulation unter Verwendung einer realen Rumpfgeometrie, dem Voith Wassertrecker (VWT). Instantanaufnahmen des HSVA Modellversuchs aus verschiedenen Blickwinkeln sind in Abb. 66 gegeben.

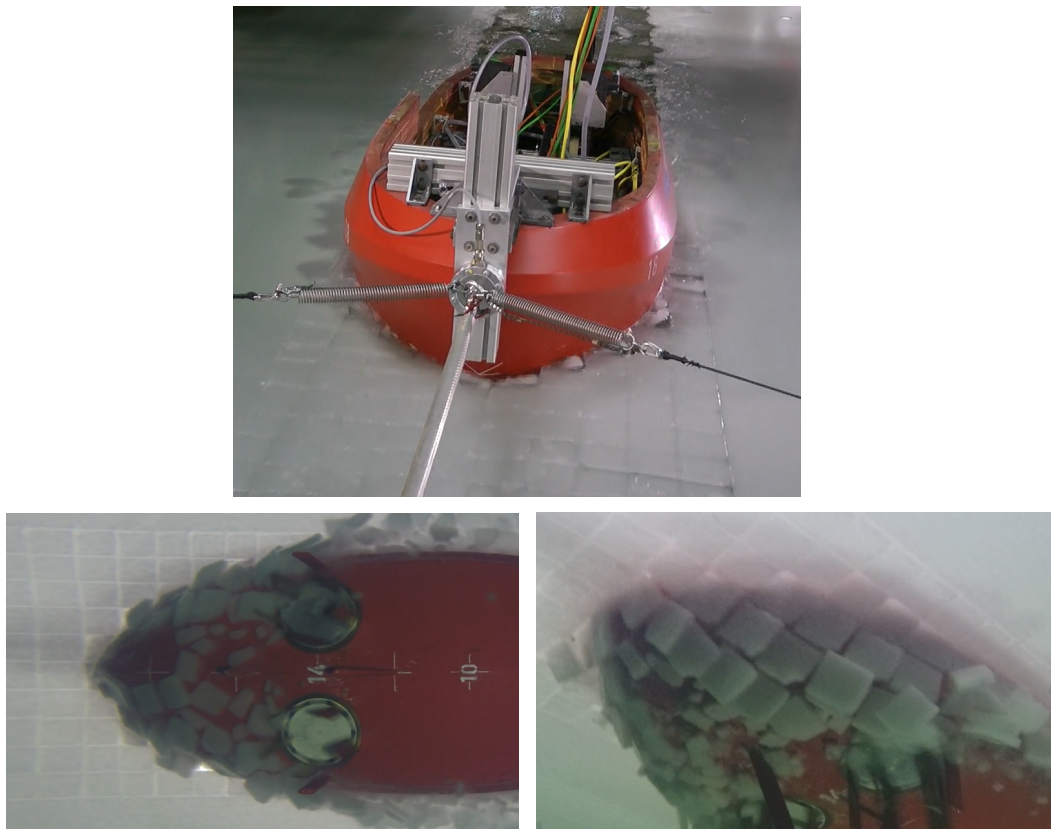


Abbildung 66: Voith Wassertrecker (VWT) Schiffsmodell mit zwei VSP Radkörpern im HSVA Eistank unter *Presawn* Eisbedingungen (10 sich über die Schiffsbreite erstreckende vorgesägte Eisschollenreihen). Frontansicht über der Wasseroberfläche (oben), Ansicht des Modells bis Mittschiff von unten (unten links) sowie Bugansicht von unten als Detail (unten rechts).

Die Abmessungen der Großausführung betragen $LOA_{GA} = 40,00\text{ m}$, $B_{GA} = 14,00\text{ m}$ bei einem Tiefgang von $CWL_{GA} = T_{GA} = 3,75\text{ m}$. Der für den HSVA Modellversuch verwendete Maßstabsfaktor beträgt $\lambda = 8 = L_{GA}/L_M$, womit sich für das Schiffsmodell die Längenmaße zu $LOA_M = 5,00\text{ m}$, $B_M = 1,75\text{ m}$ und $CWL_M = T = 0,47\text{ m}$ ergeben.

In Abb. 67 ist eine Instantanaufnahme des **elbe** Simulationsergebnisses für die zum HSVA Modellversuch äquivalente Parameterwahl dargestellt. Der simulierte Betriebspunkt entspricht einem Fortschrittsgrad von $J = 0.119$, bei einer homogenen Anströmgeschwindigkeit von $v_{inf} = 0,45\text{ m/s}$ und einer VSP Drehzahl von $n = 3\text{ Hz}$. Die Eisbedingungen werden durch Schollen der Kantenlänge $s_{Eis} = 0,175\text{ m}$ sowie einer Eisdicke von $h_{Eis} = 0,046\text{ m}$ charakterisiert. Im Gegensatz zu der detaillierten Validierung anhand von Histogrammen zur Beschreibung der Schub- und Momentenschwankungen der Operation in Eisbedingungen erfolgt die Analyse in diesem Abschnitt lediglich auf Basis einer qualitativen Beurteilung anhand von Instantanaufnahmen und dient dem Ausblick auf umfangreichere Simulationen realer Schiffsgeometrien.

Bezogen auf die Simulationen des generischen Rumpfes – mit einer Initialisierung der Eisschollen mittschiffs unter dem Rumpf und der Simulation des Hinterschiffes – wird für die VWT Geometrie eine Initialisierung der Eisschollen in der Nähe der Wasseroberfläche verwendet (ohne jedoch die freie Oberfläche in die Simulation einzubeziehen). Das Rechengebiet umfasst hierbei den Vorschiffsbereich. Für diese Simulation wird das Abtauchen sowie das Entlanggleiten der Schollen am Rumpf aufgrund der Initialisierung in Wasserlinienhöhe eingeschlossen.

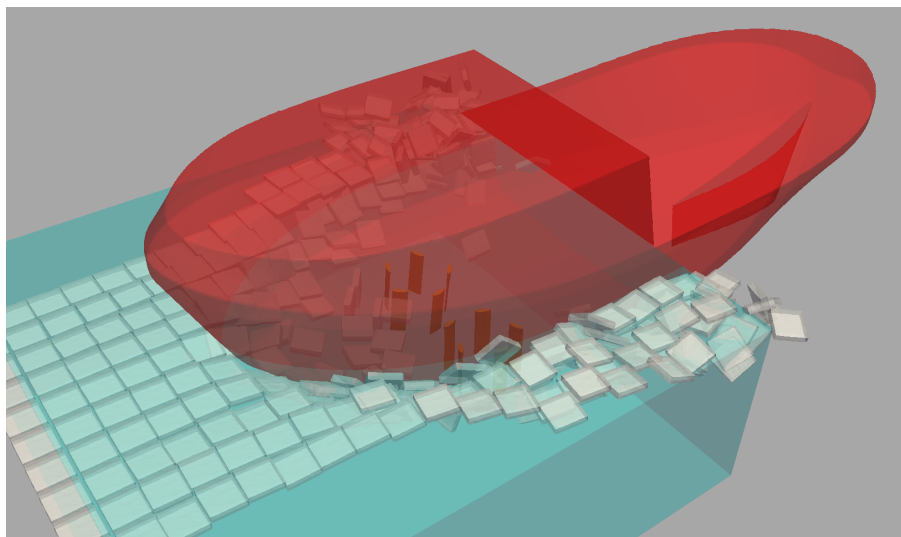


Abbildung 67: Voith Wassertrecker Rumpfgeometrie mit zwei VSP Radkörpern in Interaktion mit Eis. Die vorgebrochenen Eisschollen werden in der Nähe der freien Wasseroberfläche in das Rechengebiet eingeführt und bewegen sich anschließend mit der homogenen Anströmgeschwindigkeit auf den Rumpf zu. Nach dem Abtauchen und Entlanggleiten am Rumpf treten die Schollen in Interaktion mit den VSP Profilen. In der vorliegenden Konfiguration ($v_{inf} = 0,45\text{ m/s}$, $n = 3\text{ Hz}$) bewegen sich 10 Schollenreihen (Kantenlänge $s_{Eis} = 0,175\text{ m}$) auf den Schiffsrumpf zu. Bei Sättigung befinden sich ca. 280 Eisschollen im Rechengebiet, die sowohl untereinander als auch mit dem Rumpf und den VSP Profilen in Interaktion stehen.

Für den simulierten Betriebspunkt und die definierten Eisbedingungen entspricht die Schollenanzahl bei Sättigung im Rechengebiet ca. 280. Die Simulationszeit von 30s Echtzeit auf einer Nvidia K40m betrug 45 Stunden, wobei die Performance bei der erreichten Sättigung und der vollen Interaktion der Schollen untereinander, der Schollen mit dem Rumpf sowie den Propulsoren in etwa 60 MNUPS (engl. *Million Node Updates Per Second*, Millionen Knotenaktualisierungen pro Sekunde) betrug. Für die Simulation wurde eine Diskretisierung mit insgesamt ca. 36.8 Millionen Knoten herangezogen und es wurde eine alle 10 VSP Flügel umfassende Gitterverfeinerungsregion verwendet. Daraus resultiert eine Gitterweite von 4mm im feinen Gitter, womit die Sehnenrichtung des Profils am Schaftende mit 28 diskreten Knoten aufgelöst wird.

Der Vergleich des Eisschollenflusses zwischen den Abbildungen Abb. 66 (HSVA Modellversuch) und Abb. 67 (elbe Simulationsergebnis) weist eine gute Übereinstimmung der Dynamik auf.

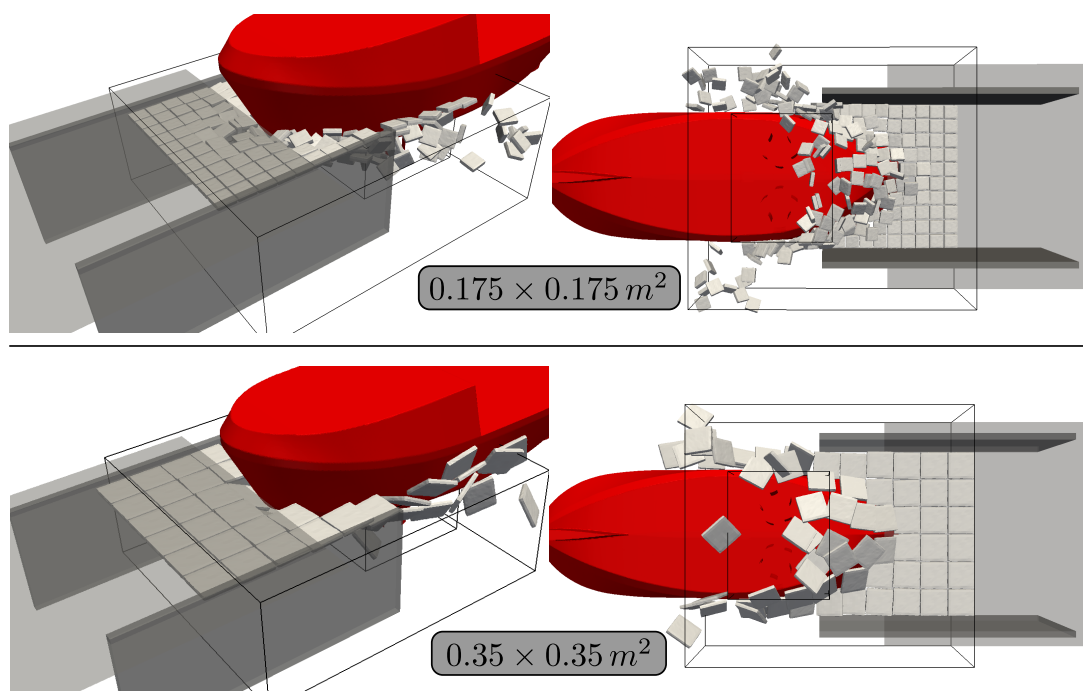


Abbildung 68: Instantanaufnahmen von Simulationen des Voith Wassertreckers unter Eisbedingungen für reale Parameterkombinationen (homogene Anströmgeschwindigkeit des Rumpfes, VSP Drehzahl, Bedeckungsgrad, usw.) gemäß den im Eistank der HSVA ausgeführten Modellversuchen. Simulation des Schollenflusses entlang des Rumpfes eines Schiffes, das in einer mit vorgebrochenen Eisschollen (definiert anhand des typischen Bruchmusters des Rumpfes) gefüllten Eisrinne hohen Bedeckungsgrades operiert. Vergleich der Scholldynamik für zwei Eisschollengeometrien unterschiedlicher Kantenlänge ($s_{\text{Eis}} = \{175; 350\}$ mm bei einer Eisdicke von jeweils $h_{\text{Eis}} = 0,046$ m) aus der Totalen (Abbn. links) und der Ansicht von unten (Abbn. rechts).

Der Schollenfluss in den Abbn. 68 der numerischen Simulationen für variierende Eisschollenabmessungen führt zu der Schlussfolgerung, dass es mittels elbe möglich ist ein physikalisch konsistentes Verhalten von Schiffs-Eis-Interaktionen zu modellieren.

14 Abschließende Bemerkungen

Das Hauptaugenmerk der in diesem Abschlussbericht dargestellten Arbeiten der Technischen Universität Hamburg (TUHH) als Beitrag zum Verbundforschungsvorhaben PROEIS lag auf einer detaillierten Untersuchung der Modellierung der Umströmung von Schiffen in eisbedeckten Gewässern mit einem hochaufgelösten, nicht-linearen Strömungssimulationsverfahren in einer bidirektionalen, monolithischen Kopplung mit einem Starrkörperlöser. Die Arbeiten des Teilprojektes der TUHH mündeten in der Entwicklung und Anwendung eines komplexen numerischen Eistanks mittels dessen die Simulation der Umströmung von Schiffsrümpfen mit einer Anzahl der Größenordnung $\mathcal{O}(10^3)$ Eisschollen möglich ist. Die Eisschollen bewegen sich dabei voll-dynamisch in dem Rechengebiet und stehen sowohl untereinander als auch mit dem Rumpf und den Propulsionsorganen sowie dem umgebenen Fluid in Interaktion. Die genaue Analyse der Scholldynamik im Vergleich zu den im Eistank der HSVA durchgeführten Modellversuchen für unterschiedliche Rümpfe wie auch verschiedene Propulsionsarten (Azi-Pod Antrieb und Voith-Schneider-Propeller) zeigte eine gute Übereinstimmung, womit die physikalische Konsistenz des entwickelten Strömungslösers nachgewiesen werden konnte.

Das Hauptaugenmerk der Umsetzung des numerischen Eistanks lag dabei in dessen hocheffizienter Implementierung des komplexen Zusammenspiels vieler involvierter Funktionalitäten und Module (Eisschollengenerator, lokale Gitterverfeinerungen, Abbildung der Propulsorkinematik, Aufprägen von Randbedingungen, Lastintegration auf Körper, Messmethoden, etc.). In diesem Kontext wurden zur Effizienzsteigerung vorzugsweise alle Methoden und Module zur Parallelisierung auf Grafikkarten als GPU-Kernel implementiert, um diese massiv parallel abzuarbeiten und dadurch kurze Durchsatzzeiten zu ermöglichen. Durch die tiefgreifenden Optimierungen sowie die Ausnutzung der verfügbaren Hardware-Ressourcen ist es gelungen einen numerischen Eistank für die Simulation von anwendungsorientierten Eistank-Versuchskonfigurationen zu etablieren, der Rechenzeiten im Bereich von Tagen anstatt von Monaten im Vergleich zu konventionellen Methoden (partitionierte Ansätze) und Lösern (RANS) ermöglicht und aus Sicht eines Ingenieurs mit einem vertretbaren Zeitaufwand bzw. tragbaren Antwortzeiten einhergeht.


Im Rahmen des Forschungsprojektes wurden zudem maßgeschneiderte umfangreiche Messmethoden zur Analyse der Propulsorbeeinflussung durch Eis entwickelt. Die Validierung anhand verfügbarer und zur Laufzeit des Projektes erhobener Referenzdaten, basierend auf den von der HSVA durchgeführten Modellversuchen, ergab, dass es mittels der entwickelten Simulationsumgebung `eIbe` in dem aktuellen Umfang möglich ist die Eis-Rumpf- sowie Eis-Propulsor-Interaktionen – unter Verwendung einiger wohldefinierter Modellbildungen – hinreichend genau numerisch abzubilden. Arbeiten die weit über den Umfang des Projektes hinausgehen – aber deren abzubildende Effekte in allen untersuchten Modellversuchen zu beobachten sind – stellen die umfangreiche Implementierung eines spanenden Ansatzes (extrem feines *Milling*) voraus, der eine sehr feine Zerteilung des Eises am Propulsor berücksichtigt und somit zu einer erheblichen Anzahl an Körpern $\approx \mathcal{O}(10^6)$ führt, die damit bedeutend größer ist als jene der im Kontext des TUHH Teilvorhabens ausgeführten Simulationen.

Literatur

- [1] Google group: ode-users. <https://groups.google.com/d/forum/ode-users>.
- [2] Jiglibx physic library. <https://jiglibx.codeplex.com>.
- [3] The ode archives. http://www.ode.org/old_list_archives.
- [4] Open dynamics engine (ode) community wiki. <http://ode-wiki.org>.
- [5] Opentissue. <http://www.opentissue.org>.
- [6] Siconos: A software for modeling and simulation of nonsmooth dynamical systems. <http://siconos.gforge.inria.fr>.
- [7] True axis physics sdk. <http://trueaxis.com/physics>.
- [8] Algoryx. Agx dynamics. <http://www.algoryx.se>.
- [9] M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, 1997.
- [10] G. Arechavaleta, E. López-Damian, and J. L. Morales. On the use of iterative lcp solvers for dry frictional contacts in grasping. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. IEEE, 2009.
- [11] M. Ashley-Rollman. Parallelizing ode. <https://www.cs.cmu.edu/~mpa/ode/>.
- [12] P. Asinari. Asymptotic analysis of multiple-relaxation-time lattice Boltzmann schemes for mixture modeling. *Computers & Mathematics with Applications*, 55(7):1392–1407, 2008.
- [13] H. Asmuth. Development of Overset Strategies for LBM-Based Flow Solvers. Masterarbeit, Technische Universität Hamburg-Harburg, 2016.
- [14] D. Baraff. Linear-time dynamics using lagrange multipliers. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 137–146. ACM, 1996.
- [15] P. Bhatnagar, E. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94:511–525, 1954.
- [16] J. Blazek. Chapter 11 - principles of grid generation. In J. Blazek, editor, *Computational Fluid Dynamics: Principles and Applications (Third Edition)*, pages 357 – 393. Butterworth-Heinemann, Oxford, third edition edition, 2015.

- [17] A. Boeing and T. Bräunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, pages 281–288, New York, NY, USA, 2007. ACM.
- [18] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a boltzmann-lattice fluid with boundaries. *Physics of Fluids (1994-present)*, 13(11):3452–3459, 2001.
- [19] A. Caiazzo and M. Junk. Boundary forces in lattice boltzmann: Analysis of momentum exchange algorithm. *Computers & Mathematics with Applications*, 55(7):1415–1423, 2008.
- [20] D. Chapman. Jiglib. <http://www.rowlhouse.co.uk/jiglib/>.
- [21] Y. Chen, Q. Cai, Z. Xia, M. Wang, and S. Chen. Momentum-exchange method in lattice boltzmann simulations of particle-fluid interactions. *Physical Review E*, 88(1):013303, 2013.
- [22] CM LABS. Vortex dynamics. <http://www.cm-labs.com>.
- [23] R. Cottle. *The linear complementarity problem*. Academic Press, Boston, 1992.
- [24] R. W. Cottle and G. B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and its Applications*, 1(1):103–125, 1968.
- [25] E. Coumans. Bullet physics library, 2015. <http://www.bulletphysics.org>.
- [26] D. B. Kirk, W. W. Hwu. *Programming Massively Parallel Processors - A Hands-on Approach*. Morgan Kaufmann, 2010.
- [27] C. Daley, S. Alawneh, D. Peters, B. Quinton, and B. Colbourne. GPU Modelling of Ship Operations in Pack Ice. *CETECH12-109-RF*, 2012.
- [28] C. Daley, S. Alawneh, D. Peters, B. Quinton, and B. Colbourne. GPU-Event-Mechanics of Ice Impact Load Statistics. *Offshore Technology Conference Paper 24645*, 2014.
- [29] P. J. Dellar. Nonhydrodynamic modes and a priori construction of shallow water lattice Boltzmann equations. *Physical Review E*, 65(3):036309, 2002.
- [30] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L. S. Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Phil. Trans. R. Soc. A*, 360:437–451, 2002.
- [31] E. Drumwright, J. Hsu, N. Koenig, and D. Shell. Extending open dynamics engine for robotics simulation. In N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, and O. von Stryk, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, volume 6472 of *Lecture Notes in Computer Science*, pages 38–50. Springer Berlin Heidelberg, 2010.

- [32] E. Kandrot, J. Sanders. *CUDA BY EXAMPLE - An Introduction to General-Purpose GPU Programming*. Addison-Wesley, 2010.
- [33] K. Erleben. *Stable, robust, and versatile multibody dynamics animation*. PhD thesis, University of Copenhagen, 2004.
- [34] esa. Polar ice cover. URL http://www.esa.int/spaceinimages/Images/2005/08/Polar_ice_cover.
- [35] M. Gehrke. Weiterentwicklung und Validierung eines GPU-beschleunigten, nicht-uniformen Gitter-Boltzmann-Verfahrens für turbulente Strömungen. Masterarbeit, Technische Universität Hamburg-Harburg, 2015.
- [36] S. Geller. *Ein explizites Modell für die Fluid-Struktur-Interaktion basierend auf LBM und p-FEM*. PhD thesis, Technischen Universität Carolo-Wilhelmina zu Braunschweig, 2010.
- [37] S. Geller, C. Janssen, and M. Krafczyk. A lattice boltzmann approach for distributed three-dimensional fluid-structure interaction. *Progress in Computational Physics Volume 3: Novel Trends in Lattice-Boltzmann Methods*, 3:199–216, 2013.
- [38] I. Ginzburg and K. Steiner. Lattice boltzmann model for free-surface flow and its application to filling process in casting. *Journal of Computational Physics*, 185(1):61–99, 2003.
- [39] I. Ginzburg, F. Verhaeghe, and D. d’Humières. Study of simple hydrodynamic solutions with the two-relaxation-times lattice Boltzmann scheme. *Communications in computational physics*, 3(3):519–581, 2008.
- [40] I. Ginzburg, F. Verhaeghe, and D. d’Humieres. Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions. *Commun. Comput. Phys*, 3:427–478, 2008.
- [41] C. Glocker. Simulation von harten kontakten mit reibung – eine iterative projektionsmethode. In *Schwingungen in Antrieben*, number VDI-Berichte Nr. 1968, pages 19 – 44. VDI-Verlag, Düsseldorf, 2006.
- [42] J. Götz, C. Feichtinger, K. Iglberger, S. Donath, and U. Rüde. Large scale simulation of fluid structure interaction using lattice boltzmann methods and thephysics engine. *AN-ZIAM Journal*, 50:166–188, 2008.
- [43] A. K. Gunstensen, D. H. Rothman, S. Zaleski, and G. Zanetti. Lattice boltzmann model of immiscible fluids. *Physical Review A*, 43(8):4320–4327, 1991.
- [44] F. Güttler. Realitätsnahe simulationsumgebung einer selbstorganisierenden roboterwelt. Master’s thesis, Universität Leipzig, 2007.

- [45] M. Harris. Optimizing parallel reduction in cuda. [online], http://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86_website/projects/reduction/doc/reduction.pdf, 01 2015.
- [46] Havok. Havok physics. <http://havok.com/products/physics>.
- [47] J. Hummel, R. Wolff, T. Stein, A. Gerndt, and T. Kuhlen. An evaluation of open source physics engines for use in virtual reality assembly simulations. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, editors, *Advances in Visual Computing*, volume 7432 of *Lecture Notes in Computer Science*, pages 346–357. Springer Berlin Heidelberg, 2012.
- [48] K. Iglberger and U. Rüde. Massively parallel rigid body dynamics simulations. *Computer Science - Research and Development*, 23(3-4):159–167, 2009.
- [49] K. Iglberger and U. Rüde. The pe rigid multi-body physics engine. Technical report, University of Erlangen-Nürnberg, 2009.
- [50] S. Ivaldi, V. Padois, and F. Nori. Tools for dynamics simulation of robots: a survey based on user feedback. *CoRR*, abs/1402.7050, 2014.
- [51] J. Tölke. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. *Computing and Visualization in Science*, 13(1):29–39, 2010.
- [52] C. F. Janßen, D. Mierke, M. Übrück, S. Gralher, and T. Rung. Validation of the GPU-Accelerated CFD Solver ELBE for Free Surface Flow Problems in Civil and Environmental Engineering. *Computation*, 3:354–385, 2015.
- [53] C. F. Janßen. *Kinetic approaches for the simulation of non-linear free surface flow problems in civil and environmental engineering*. PhD thesis, Technischen Universität Carolo-Wilhelmina zu Braunschweig, 2010.
- [54] C. F. Janßen, N. Koliha, and T. Rung. A fast and rigorously parallel surface voxelization technique for gpu-accelerated cfd simulations. *Communications in Computational Physics*, 17:1246–1270, 5 2015.
- [55] C. F. Janßen and M. Krafczyk. Free surface flow simulations on GPGPUs using the LBM. *Computers & Mathematics with Applications*, 61(12):3549–3563, 2011. International Conferences on Mesoscopic Methods in Engineering and Science.
- [56] J. Jerez and A. Suero. Newton game dynamics. <http://newtondynamics.com>.
- [57] jmil. High resolution stanford bunny. [online], verfügbar als STL-Datei unter  auf <http://www.thingiverse.com/thing:11622>, 09 2015.

- [58] M. Jussila and H. Soininen. Interaction between Ice and Propeller. *Research Reports 1281 - Technical Research of Finland, Helsinki University of Technology*, 1991.
- [59] R. Klein. *Algorithmische Geometrie*. Springer Berlin Heidelberg, 2005.
- [60] S. Kollmannsberger, S. Geller, A. Düster, J. Tölke, C. Sorger, M. Krafczyk, and E. Rank. Fixed-grid fluid-structure interaction in two dimensions based on a partitioned lattice boltzmann andp-FEM approach. *International Journal for Numerical Methods in Engineering*, 79(7):817–845, aug 2009.
- [61] R. Kooijman. Evaluation of open dynamics engine software. Technical Report 2010.022, Eindhoven University of Technology, Department of Mechanical Engineering, Dynamics & Control, 2010.
- [62] C. Körner, M. Thies, T. Hofmann, N. Thürey, and U. Rüde. Lattice boltzmann model for free surface flow for modeling foaming. *Journal of Statistical Physics*, 121(1-2):179–196, 2005.
- [63] P. Lallemand and L.-S. Luo. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics*, 184:406–421, 2003.
- [64] D. Lam. Tokamak physics engine. <http://www.tokamakphysics.com>.
- [65] S.-K. Lee. Rational Approaches to Integrate the Design of Propulsion Power and Propeller Strength for Ice Ships. *ABS Technical Paper*, 2006.
- [66] F. Leon. Gimpact – geometric tools for vr. [online], <http://gimpact.sourceforge.net>, 10 2014.
- [67] R. Lubbad and S. Loiset. A numerical model for real-time simulation of ship-ice interaction. *Cold Regions Science and Technology*, 65:111–127, 2011.
- [68] M. Krafczyk. *Gitter-Boltzmann-Methoden: Von der Theorie zur Anwendung*. Habilitation, Technische Universität München, 2001.
- [69] R. Mei, D. Yu, W. Shyy, and L.-S. Luo. Force evaluation in the lattice boltzmann method involving curved geometry. *Physical Review E*, 65(4):041203, 2002.
- [70] D. Merrill. Cub – cuda unbound. [online], <https://nvlabs.github.io/cub>, 01 2015.
- [71] I. Metrikin, A. Borzov, R. Lubbad, and S. Løset. Numerical simulation of a floater in a broken-ice field: Part ii – comparative study of physics engines. In *Proceedings of the ASME 2012 31st International Conference on Ocean, Offshore and Arctic Engineering (OMAE2012)*, volume 6, pages 477–486, 2012.

- [72] I. Metrikin, A. Borzov, R. Lubbad, and S. Loset. Numerical Simulation of Floater in a Broken Ice Field – Part II: Comparative Study of Physics Engines. *OMAE Paper*, 83430, 2012.
- [73] I. Metrikin, S. Loset, N. Jenssen, and S. Kerkeni. Numerical Simulation of Dynamic Positioning on Ice. *Marine Technology Society Journal*, 47:14–30, 2013.
- [74] I. Metrikin, W. Lu, R. Lubbad, S. Løset, and M. Kashafutdinov. Numerical simulation of a floater in a broken-ice field: Part i – model description. In *Proceedings of the ASME 2012 31st International Conference on Ocean, Offshore and Arctic Engineering (OMAE2012)*, volume 6, pages 579–588, 2012.
- [75] D. Mierke, C. Janßen, and T. Rung. An efficient algorithm for the calculation of sub-grid distances for higher-order lbm boundary conditions in a gpu simulation environment. *Computers & Mathematics with Applications*, 2018.
- [76] I. Muggeridge and A. Aboulazm. Analytical investigation of ship resistance in broken or pack ice. *OMAE Conf. Proc.*, 4:359–365, 1989.
- [77] J. Möbius. Openmesh – a generic and efficient polygon mesh data structure, 2018. <https://www.openmesh.org>.
- [78] L. Nasdala. *FEM-Formelsammlung Statik und Dynamik*. Springer-Verlag, Berlin Heidelberg New York, 2012.
- [79] N.-Q. Nguyen and A. J. Ladd. Sedimentation of hard-sphere suspensions at low reynolds number. *Journal of Fluid Mechanics*, 525:73–104, 2005.
- [80] nVIDIA. nVIDIA CUDA. URL http://www.nvidia.com/object/cuda_home_new.html.
- [81] Nvidia. Nvidia physx. <http://www.geforce.com/hardware/technology/physx>.
- [82] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [83] Y. Qian, D. d’Humières, and P. Lallemand. Lattice BGK models for Navier Stokes equation. *Europhysical Letters*, 17:479–784, 1992.
- [84] J. Reinders. *Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism*. O’Reilly Series. O’Reilly Media, 2007.
- [85] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th Edition*. Kitware, 2006.
- [86] A. Seugling and M. Rölin. Evaluation of physics engines and implementation of a physics module in a 3d-authoring tool. Master’s thesis, UmeåUniversity, Department of Computing Science, 2006.

- [87] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 91:99–164, 1963.
- [88] R. Smith. Ode – open dynamics engine. [online], <http://www.ode.org>, 10 2014.
- [89] R. Smith. Open dynamics engine, v0.5 user guide. [online], <http://ode.org/ode-latest-userguide.pdf>, 10 2014.
- [90] Soininen, H. *A Propeller-Ice Contact Model*. Dissertation, Helsinki University of Technology, 1998.
- [91] P. Song, J.-S. Pang, and V. Kumar. A semi-implicit time-stepping model for frictional compliant contact problems. *International Journal for Numerical Methods in Engineering*, 60(13):2231–2261, 2004.
- [92] J. Sreeram and S. Pande. Parallelizing a real-time physics engine using transactional memory. In E. Jeannot, R. Namyst, and J. Roman, editors, *Euro-Par 2011 Parallel Processing*, volume 6853 of *Lecture Notes in Computer Science*, pages 206–223. Springer Berlin Heidelberg, 2011.
- [93] W. Stamm. *Modellierung und Simulation von Mehrkörpersystemen mit flächigen Reibkontakten*. Schriftenreihe des Instituts für Technische Mechanik. KIT Scientific Publ., 2011.
- [94] D. E. Stewart and J. C. Trinkle. Dynamics, friction, and complementarity problems. *International Conference on Complementarity Problems*, 1995.
- [95] D. E. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996.
- [96] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110 – 136, 2003.
- [97] P. Terdiman. Sweep and prune (sap), 2007. [online], <http://www.codercorner.com/SAP.pdf>.
- [98] P. Terdiman. Opcode – optimized collision detection. [online], <http://www.codercorner.com/Opcode.htm>, 10 2014.
- [99] N. Thürey and U. Rüde. Stable free surface flows with the lattice boltzmann method on adaptively coarsened grids. *Computing and Visualization in Science*, 12(5):247–263, 2009.
- [100] J. Tölke, S. Freudiger, and M. Krafczyk. An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations. *Computers & Fluids*, 35(8):820–830, 2006.

- [101] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with coulomb friction. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 77(4):267–279, 1997.
- [102] C. D. Twigg. *Controlling Multibody Dynamics via Browsing and Time Reversal*. PhD thesis, Pittsburgh, PA, USA, 2008. AAI3314657.
- [103] M. Überrück. Alternative advektionsschemata und auslassrandbedingungen für numerische schleptanks. Masterarbeit, Technische Universität Hamburg-Harburg, 2015.
- [104] B. Veitch. Predictions of Ice Contact Forces on a Marine Screw Propeller during the Propeller-Ice Cutting Process, *Acta Polytechnica Scandinavica. Mech. Engineering Series*, 118, 1995.
- [105] Veitch, B. *Propeller-ice Interaction*. Disseration, Helsinki University of Technology, 1992.
- [106] L.-P. Wang, C. Peng, Z. Guo, and Z. Yuc. Lattice Boltzmann simulation of particle-laden turbulent channel flow. *Computers and Fluids*, 124:226–236, 2016.
- [107] Wang, J. *Prediction of Propeller Performance on a Model Podded Propulsor in Ice (Propeller-Ice-Interaction)*. Disseration, Faculty of Engineering and Applied Science, Memorial Univ. of Newfoundland, Canada, 2007.
- [108] J. Wittenburg. *Dynamics of Multibody Systems*. Springer-Verlag, 2008. ISBN 978-3-540-73913-5.
- [109] R. Woolgar and D. Colbourne. Effect of hull-ice friction coefficient on prediction of pack ice forces for moored offshore vessels. *Ocean Engineering*, 37:296–302, 2010.
- [110] D. Yu, R. Mei, L. S. Luo, and W. Shyy. Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences*, 39:329–367, 2003.
- [111] D. Yu, R. Mei, L.-S. Luo, and W. Shyy. Viscous flow computations with the method of lattice boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367, 2003.

Berichtsblatt

1. ISBN oder ISSN geplant	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel MOUSE – Modellierung der Umströmung von Schiffen in Eisbedeckten Gewässern	
4. Autor(en) [Name(n), Vorname(n)] Mierke, Dennis Gehrke, Martin Janßen, Christian Friedrich Rung, Thomas	5. Abschlussdatum des Vorhabens 30.09.2018
	6. Veröffentlichungsdatum Februar 2019
	7. Form der Publikation Bericht
8. Durchführende Institution(en) (Name, Adresse) Technische Universität Hamburg Institut für Fluidodynamik und Schiffstheorie Am Schwarzenberg-Campus 4 21073 Hamburg	9. Ber. Nr. Durchführende Institution FDS/002-2019/ProEis
	10. Förderkennzeichen 03SX391G
	11. Seitenzahl 112
12. Fördernde Institution (Name, Adresse) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. Literaturangaben 111
	14. Tabellen 4
	15. Abbildungen 68
16. Zusätzliche Angaben Keine	
17. Vorgelegt bei (Titel, Ort, Datum) Projekträger Jülich (PTJ) – Maritime Technologien (MGS2), 10969 Berlin, Herr Daniel Testa; Februar 2019.	
18. Kurzfassung Das Hauptaugenmerk der in diesem Abschlussbericht dargestellten Arbeiten der Technischen Universität Hamburg (TUHH) als Beitrag zum Verbundforschungsvorhaben <i>ProEis</i> lag auf einer detaillierten Untersuchung der Modellierung der Umströmung von Schiffen in eisbedeckten Gewässern mit einem hochaufgelösten, nicht-linearen Strömungssimulationsverfahren in einer bidirektionalen, monolithischen Kopplung mit einem Kontaktdynamik-Simulationstool. Die Entwicklungsarbeiten des Teilprojektes der TUHH mündeten in der Entwicklung und Anwendung eines komplexen numerischen Eistanks mittels dessen die Simulation der Umströmung von Schiffsrümpfen mit einer Anzahl in der Größenordnung von 1000 Eisschollen möglich ist. Die Eisschollen bewegen sich dabei voll-dynamisch in dem Rechengebiet und stehen sowohl untereinander als auch mit dem Rumpf und den Propulsionsorganen in Interaktion. Die genaue Analyse der Scholldynamik im Vergleich zu den im Eistank der HSVA durchgeführten Modellversuche für unterschiedliche Rümpfe wie auch verschiedene Propulsionsarten (Azi-Pod Antrieb und Voith-Schneider-Propeller) zeigte eine gute Übereinstimmung, womit die physikalische Konsistenz des entwickelten der Strömungslösers nachgewiesen werden konnte.	
19. Schlagwörter LBM, GPU, CUDA, Schiff-Eis-Interaktion, Numerischer Eistank, Voith-Schneider-Propeller	
20. Verlag	21. Preis

Document Control Sheet

1. ISBN or ISSN planned	2. type of document (e.g. report, publication) Final Report
3. title MOUSE – Modellierung der Umströmung von Schiffen in Eisbedeckten Gewässern	
4. author(s) (family name, first name(s)) Mierke, Dennis Gehrke, Martin Janßen, Christian Friedrich Rung, Thomas	5. end of project 30.09.2018
	6. publication date February 2019
	7. form of publication periodical
8. performing organization(s) (name, address) Hamburg University of Technology Institute for Fluid Dynamics und Ship Theory Am Schwarzenberg-Campus 4 21073 Hamburg	9. originator's report no. FDS/002-2019/ProEis
	10. reference no. 03SX391G
	11. no. of pages 112
12. sponsoring agency (name, address) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. no. of references 111
	14. no. of tables 4
	15. no. of figures 68
16. supplementary notes None	
17. presented at (title, place, date) Project Office Jülich (PTJ) – Maritime Technologies (MGS2), 10969 Berlin, Germany; Mr. Daniel Testa; February 2019	
18. abstract The main focus of the work at Technical University of Hamburg (TUHH) contributing to the joint collaborative research project <i>ProEis</i> was a numerical model for the flow around ships in ice-covered areas. The procedure consist of a viscous (turbulent) flow solver with a bidirectional, monolithic coupling to a contact-dynamics simulation procedure. The outcome of the project is the development and application of a complex numerical ice tank for the simulation of the flow around ship hulls in pack-ice conditions with up to approximately 1000 ice floes. The ice floes move fully-dynamically within the computational domain and interact with each other as well as with the hull and the propulsion devices. The analysis of the ice floe dynamics in model scale reveals a fair agreement with experimental results carried out at an ice tank for different hulls as well as different types of propulsors (Azi-Pod and Voith-Schneider), which proves the physical consistency of the computational framework.	
19. keywords LBM, GPU, CUDA, Ship-Ice-Interation, Numerical Ice Tank, Voith-Schneider-Propeller	
20. publisher	21. price