

Numerische Erzeugung von Wasserwellen zur Analyse von Wellenkraftwerken

Christina Peuker

Geboren am 23. Mai 1989 in Bergisch Gladbach

29. August 2011

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Michael Griebel

INSTITUT FÜR NUMERISCHE SIMULATION

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER
RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

Inhaltsverzeichnis

1	Einleitung	1
2	Wellenkraft und Numerische Wellenbecken	5
2.1	Wie entstehen Wasserwellen?	5
2.2	Wellenkraftwerke	7
2.2.1	Pelamis	8
2.2.2	Oscillating–Water–Column	8
2.3	Numerische Wellenbecken	10
2.4	Wellenerzeuger Theorie	10
3	Strömungssimulation mit NaSt3DGPF	12
3.1	Die Navier–Stokes–Gleichungen	12
3.2	Zweiphasenströmung	12
3.3	Randbedingungen	13
3.4	Die Level–Set Methode	14
3.5	Diskretisierung des Zweiphasenmodells	16
3.5.1	Das Gitter	16
3.5.2	Die Level–Set Funktion	16
3.5.3	Reinitialisierung und Masseerhaltung	17
3.5.4	Die Chorin–Temam–Projektionsmethode	18
3.6	Parallelisierung	18
3.7	Der Algorithmus zum Lösen des Zweiphasensystems	19
4	Numerische Wellenerzeugung	20
4.1	Wellenerzeugung am Computer	20
4.2	Der Versuchsaufbau	21
4.3	Starrkörper Darstellungen auf dem Gitter	22
4.4	Komplexe Starrkörper	22
4.5	Fluid–Struktur–Interaktion	23
4.5.1	Die Modellgleichungen	23
4.5.2	Einbindung des Starrkörpers in das Zweiphasenmodell	24
4.5.3	Der erweiterte Algorithmus	25
4.6	Der Algorithmus erweitert auf die Interaktion mit Starrkörpergeometrien	27
5	Simulationsergebnisse	28
5.1	Realisierung von Numerischen Wellenbecken	28
5.1.1	Das Becken	28
5.1.2	Die Wellenerzeuger	29
5.1.3	Die Kraftwerke	29
5.2	Numerische Simulationen zur Wellenerzeugung	30
5.3	Numerische Simulationen zum Testen von Wellenkraftwerken	33
6	Zusammenfassung und Ausblick	35
	Anhang	36
	Literatur	44

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die zum Entstehen dieser Arbeit beigetragen haben. Ich bedanke mich bei Herrn Prof. Dr. Griebel für die Vergabe dieses äußerst spannenden Themas sowie viele hilfreiche Denkanstöße und Anregungen. Außerdem danke ich allen Mitarbeitern des Instituts für numerische Simulation, die einem zu jeder Zeit mit Rat und Tat zur Seite stehen. Insbesondere gilt mein Dank meinem Betreuer Peter Zaspel, der mir mit seinen zahlreichen Anregungen und Korrekturen enorm weitergeholfen hat. Weiterhin möchte ich mich bei Markus Burkow für die Motivation und einige interessante Ideen bedanken. Ein weiteres Dankeschön an Ralph Thesen für die technische Unterstützung.

Für Korrekturlesen und ständige Motivation danke ich außerdem Laura, Thomas und meinen Eltern. Vor allem möchte ich mich bei meiner Familie für ihr Verständnis und die enorme Unterstützung bedanken.

1 Einleitung

Das Meer birgt eine große Menge an bisher weitgehend ungenutzter Energie. Seit über hundert Jahren beschäftigen sich Wissenschaftler bereits mit dem Thema, diese Energie mit Hilfe von Wellenkraftwerken nutzbar zu machen und beispielsweise in Strom umzuwandeln. Dabei ist bereits eine große Anzahl verschiedenster Ansätze entstanden, die in Laborversuchen und auch in realer Größe getestet werden müssen. Ein enormes Problem stellen dabei die Kosten und der Aufwand dar, diese Prototypen unter den Gegebenheiten in der Natur zu untersuchen.

Eine Lösung dafür findet sich in Computersimulationen, genauer gesagt in der Computer gestützten Strömungsdynamik (*CFD* — *Computational Fluid Dynamics*). Damit lassen sich kleinere Laborversuche ebenso wie großskalige Projekte simulieren, ohne die Wellenkraftwerke tatsächlich bauen zu müssen.

Eben solche Simulationen sind nun Ziel dieser Arbeit. Die erste Aufgabe besteht darin, die in der Natur vorkommenden Wasserwellen möglichst realistisch virtuell zu erzeugen. Wurde einmal gezeigt, dass jede Art von Welle generiert werden kann, so können auch Wellenkraftwerke unter beliebigen Bedingungen getestet werden.

Motivation

Ideen für Wellenkraftwerke gibt es viele. Das Umsetzen der Idee zu einem Prototypen oder gar einem fertigen Produkt, das kommerziell eingesetzt werden kann, scheidet oftmals an den finanziellen Mitteln. Die Strömungsdynamik bietet eine gute Möglichkeit, die Kraftwerke zu testen, ohne diese kostspielig an geeigneten Standorten installieren zu müssen, wo diese jeglichen Naturgewalten ausgesetzt sind. Beispielsweise wurde 1985 auf der norwegischen Insel Toftestallen ein Kraftwerk von der Firma Kvaerner zu Testzwecken installiert, welches bereits Ende 1988 in einem Sturm zerstört wurde [Gra95]. Andere Testbauten wurden sogar bereits während der Montage ruiniert, wie zum Beispiel das Projekt *OSPREY* (*Ocean Swell Powered Renewable Energy*), das 1995 vor der schottischen Küste in Betrieb gehen sollte [Gra95].

Beide genannten Bauwerke gehören zu den sogenannten *Oscillation Water Column (OWC)* Kraftwerken. Die Funktionsweise ist dabei prinzipiell sehr einfach und in Abbildung 1.0.1 veranschaulicht. Das Gebäude besteht aus einer Kammer, die unterhalb der Wasseroberfläche eine Öffnung hat, durch welche die Wellenenergie in das Kraftwerk gelangen kann. Am oberen, mit Luft gefüllten Teil der Kammer befindet sich ebenfalls eine Öffnung, durch die Luft entweichen kann. Gelangt nun eine Welle an das Bauwerk, so steigt im Inneren der Wasserspiegel und die Luft oberhalb der Wassersäule wird hinausgedrückt. Sinkt der Wasserspiegel wieder, so wird die Luft zurück in das Kraftwerk gesogen. Eben diese Luftbewegung soll nun eine Turbine antreiben, die sich vor der Öffnung befindet, durch die die Luft strömt. Die Turbine wiederum sollte möglichst gleichmäßig Strom erzeugen, um sinnvoll eingesetzt werden zu können.

Es gibt viele Varianten von diesen OWC-Kraftwerken, aber das Grundprinzip bleibt bei allen erhalten. Diese einfache Funktionsweise macht es gut möglich, die Wellenkraftwerke am Computer zu simulieren. Dadurch können theoretische Ansätze schnell verifiziert werden und auch neue Ideen ohne großen Aufwand getestet werden.

Aktuelle Entwicklung

Neben OWC-Kraftwerken gibt es noch eine Reihe anderer Kraftwerkstypen, die beispielsweise von Graw [Gra95, Gra06] näher beleuchtet werden. Jedoch sind nicht alle davon ökonomisch sinnvoll. Daher gibt es jedes Jahr etliche neue Entwicklungen, die die Energieausbeute verbessern sollen. Kommerzielle Erfolge sind jedoch bis dato kaum zu verbuchen. In der spanischen Stadt Matriku ist vor wenigen Wochen das erste gewerbliche Wellenkraftwerk in Betrieb gegangen, das 250 Haushalte mit Strom versorgen soll [Küf11]. Dies ist das erste Kraftwerk in Europa, das tatsächlich wirtschaftlich genutzt wird. Viele andere Wellenenergiewandler speisen zwar bereits Strom in das öffentliche Netz, sind aber geschäftlich noch nicht nutzbar.

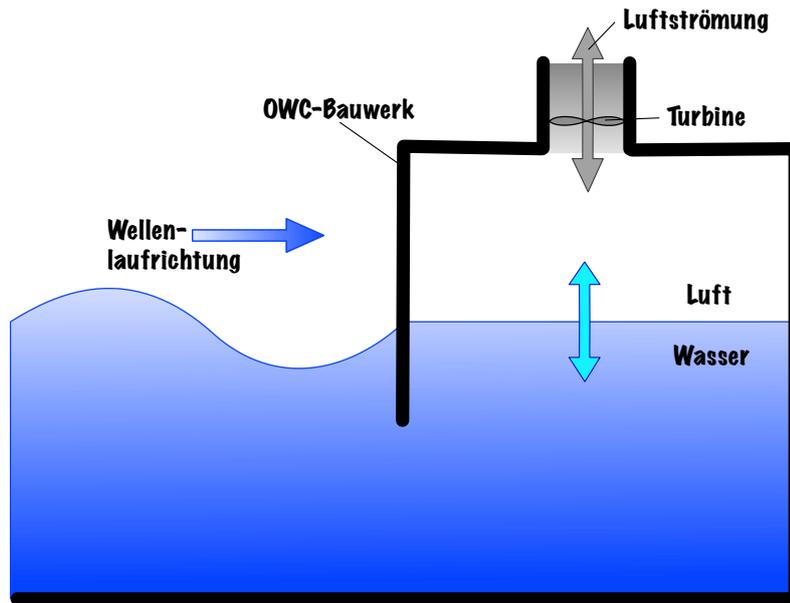


Abb. 1.0.1: Skizze eines *Oscillating-Water-Column* Kraftwerks. Die ankommenden Wellen bewirken eine vertikale Bewegung der Wassersäule im Inneren des Bauwerks. Dadurch wird die Luft im Inneren ebenfalls bewegt und treibt eine Turbine an (eigene Abbildung).

Um Wellenkraftwerke nun in Computersimulationen testen zu können, müssen zunächst einmal Wellen generiert werden. Diese Aufgabe ist keineswegs trivial. Kim et al. [KCT99] sowie Tanizawa [Tan00] geben einen guten Überblick über die verschiedenen Ansätze, Wellen in sogenannten *Numerischen Wellenbecken* zu erzeugen. Dabei unterscheiden sich die Herangehensweisen primär in der Art der Wellenerzeugung. Eine Möglichkeit besteht darin, Wellen durch Veränderungen der zugrunde liegenden Ausgangsbedingungen — die Grundgleichungen oder die Randbedingungen — zu generieren. Lin & Liu [LL99] beispielsweise verändern in einem Teil des Wellenbeckens eine der Grundgleichungen so, dass die Inkompressibilität gestört wird und es so zu Schwankungen in der Wasseroberfläche kommt. Wang et al. [WYT95] hingegen benutzen in ihrem sogenannten *LONG-TANK* modifizierte Randbedingungen, um auf diese Weise Wellen zu erzeugen. Diese Ansätze fallen in die Kategorie der *rein numerischen* Wellenerzeugung. Weitere Beispiele dafür finden sich in [CKZL99, WSH05, CY09, Sen11]. Alle diese Vorgehensweisen sind jedoch eher unphysikalisch, da sie die Ausgangsbedingungen verändern.

Ein anderes Konzept stellt die *physikalische* Wellengenerierung dar. Es handelt sich zwar weiterhin um numerische Simulationen, doch das Prinzip der Wellenerzeuger ähnelt dem Aufbau in einem Labor. Der Versuchsaufbau sieht dabei folgendermaßen aus: Es wird ein langes Wellenbecken teilweise mit Wasser gefüllt und an einem Ende ein Wellenerzeuger installiert. Dieser kann verschiedene Formen haben, folgt aber immer dem Prinzip der Wasserverdrängung. Das heißt der Wellenerzeuger wird so hin- und her beziehungsweise auf- und ab bewegt, dass das Wasser zur Seite gedrückt wird und somit Wellen entstehen. Eine ausführliche Theorie zu diesen physikalischen Wellenerzeugern finden sich in [DD90, HG09]. Am anderen Ende des Beckens befindet sich eine Art von Wellenabsorber, der eine Reflexion der Wellen verhindern soll, um Resonanzeffekte auszuschließen. Ein Beispiel für eine solche Simulation in zwei Raumdimensionen mit einem Keil als Wellenerzeuger findet sich in [Mik07].

Viele Studien versuchen die Bewegung solcher physikalischen Wellenerzeuger durch numerische Ansätze nachzuahmen; die meisten davon sind allerdings in zwei Raumdimensionen, wie beispielsweise auch [PYM⁺01]. Da das Ziel dieser Arbeit die Simulation von realistischen Hochseewellen, wie sie in der Natur vorkommen, ist, liegt eine Simulation in drei Raumrichtungen nahe. Nur so können auch dreidimensionale Kraftwerksgeometrien sinnvoll getestet werden. Einige Wellenkraftwerke nutzen beispielsweise zusätzliche Platten, welche die ankommenden Wellen bündeln, beziehungsweise die auftreffende Wellenenergie verstärken sollen. Solche Konstruktionen lassen sich in zwei Dimensionen nicht darstellen.

In vielen Wellensimulationen wird lediglich ein *einphasiges* Modell betrachtet, wie zum Beispiel in [Sen11, KfV⁺05]. Das heißt, dass nur das Wasser simuliert wird, während die Luft darüber

vernachlässigt wird. Dies ist ebenso wie bei der rein numerischen Wellenerzeugung eine unphysikalische Annahme.

Im Rahmen dieser Arbeit wird die physikalische Wellenerzeugung genutzt, um verschiedene Wellen zu generieren. Das heißt, die Wellen entstehen durch die regelmäßigen Bewegungen unterschiedlicher Festkörper in der Zweiphasenströmung.

Umsetzung

In der vorliegenden Arbeit geht es zunächst einmal um die numerische Wellenerzeugung. Es wird eine **dreidimensionale** Strömungssimulation mit **zwei Phasen** und einem **beweglichen Festkörper** durchgeführt. Das bedeutet, dass ein System mit zwei verschiedenen *Fluiden* — in diesem Fall Wasser und Luft — betrachtet wird. Außerdem spielt die Interaktion zwischen dem beweglichen Wellenerzeuger und den beiden Fluiden eine wichtige Rolle, das heißt das Modell wird um eine *Fluid-Struktur-Interaktion* erweitert. Hierbei wird die Kopplung zwischen dem Zweiphasensystem und dem Starrkörper über die *Level-Set Darstellung* der freien Oberflächen hergestellt. Soweit bekannt, wird ein solches Modell erstmalig für die Simulation von Wellen verwendet.

Mit diesem Konzept kann schließlich die Wellenerzeugung realisiert werden. Dazu werden verschiedene Arten von Wellenerzeugern erstellt, die mit verschiedenen Geschwindigkeiten getestet werden. Daraufhin können diese Ergebnisse mit der Theorie verglichen werden, um eine erfolgreiche Wellengenerierung zu demonstrieren.

Nach erfolgreicher Umsetzung, werden anschließend einige einfache Wellenkraftwerke untersucht. Dazu werden die erzeugten Wellen genutzt, um die Funktionsweise der OWC-Kraftwerke zu inspizieren. In den Simulationen wird auf die Turbinen verzichtet und lediglich der entstehende Luftstrom untersucht.

Eigene Leistung

Die Strömungssimulationen für diese Arbeit wurden mit dem Programm *NaSt3DGPF* des *Instituts für Numerische Simulation* an der Universität Bonn realisiert. Für die Erweiterung auf die Fluid-Struktur-Wechselwirkungen wurde der von Croce [Cro10] erweiterte Code genutzt. An dieser Stelle werden die wesentlichen Beiträge dieser Arbeit vorgestellt.

- Es wird ein Überblick über die Theorie der Wellenkraft und deren Nutzung, sowie die Numerische Wellenerzeugung und deren aktueller Fortschritt gegeben.
- Der bestehende Strömungslöser wird so modifiziert, dass eine Wellengenerierung mit Hilfe von Festkörpern und die Untersuchung von Wellenkraftwerken möglich ist.
- Erstmals wird ein paralleler dreidimensionaler Zweiphasen-Strömungslöser mit Fluid-Struktur-Interaktion und Level-Set Darstellung der freien Oberflächen für solche hoch genauen Simulationen erfolgreich eingesetzt, um verschiedene Arten von Wellen zu erzeugen und anschließend dreidimensionale Wellenkraftwerke zu untersuchen.

Überblick

Abschließend für dieses Kapitel soll noch eine Übersicht über die verschiedenen Teile dieser Arbeit gegeben werden.

In **Kapitel 2** wird eine kurze Einführung in die theoretischen Hintergründe zu dieser Arbeit gegeben. Dazu wird zunächst das Grundprinzip der Wellenentstehung und Wellenkraftnutzung erklärt und anschließend die Theorie zu Numerischen Wellenbecken erläutert.

Kapitel 3 behandelt die zugrunde liegenden Gleichungen der Strömungsdynamik und das Zweiphasenmodell. Dieses wird diskretisiert und parallelisiert. Zudem wird eine vollständiger Simulationsalgorithmus angegeben.

Kapitel 4 befasst sich mit der Numerischen Wellenerzeugung. Es wird der Aufbau eines Numerischen Wellenbeckens und dessen Umsetzung mit Hilfe des vorliegenden Strömungslösers erläutert. Außerdem wird die Erweiterung des Zweiphasenmodells um einen beweglichen Festkörper beschrieben und der vollständige erweiterte Algorithmus angegeben.

In **Kapitel 5** werden schließlich die Simulationsergebnisse vorgestellt und die erfolgreichen Simulationen von Wellen veranschaulicht. Außerdem werden die verschiedenen getesteten Wellenkraftwerke präsentiert.

Zu guter Letzt fasst **Kapitel 6** alle Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf denkbare Weiterentwicklungen.

2 Wellenkraft und Numerische Wellenbecken

In diesem Kapitel wird auf die Einteilung der verschiedenen Wellenarten sowie deren Entstehung in der Natur eingegangen. Weiterhin wird die Nutzung der Wellenkraft mittels Wellenkraftwerken beschrieben und an Beispielen verdeutlicht.

2.1 Wie entstehen Wasserwellen?

Wasserwellen sind Auslenkungen der Wasseroberfläche und können durch verschiedene Einflüsse, wie Gravitationskräfte, Stürme oder Erdbeben entstehen. Gezeitenwellen beispielsweise entstehen durch die Anziehungskraft von Mond und Erde und haben eine sehr lange Periode von 12 Stunden — im Gegensatz zu beispielsweise gewöhnlichen Schwerewellen, die eine Periode von einer bis 30 Sekunden haben. Die Frage ist also zunächst einmal, welche Wellen nun eigentlich für den Einsatz von Wellenkraftwerken interessant sind.

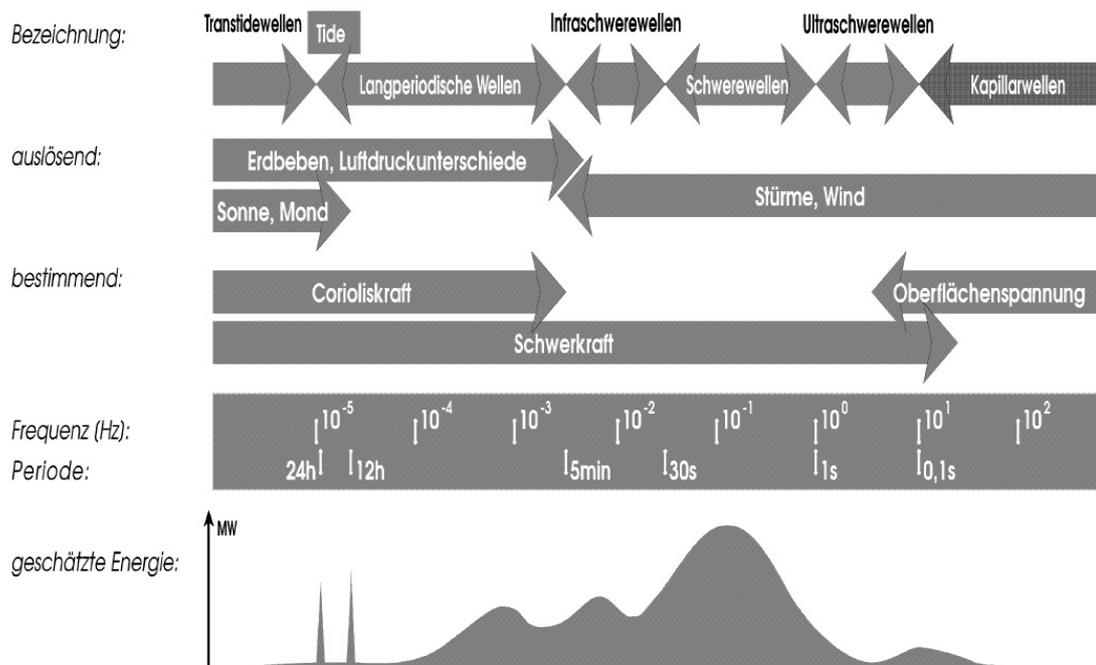


Abb. 2.1.1: Verschiedene Wellenarten eingeteilt nach Frequenz beziehungsweise Periode, den auslösenden und den bestimmenden Kräften. Außerdem ist die geschätzte Energie der verschiedenen Wellen dargestellt (aus [Gra95]).

Abbildung 2.1.1 zeigt eine Einteilung der verschiedenen Wellenarten nach ihrer Frequenz beziehungsweise Periode und ihrer Entstehung. Außerdem ist die geschätzte Energie abgebildet und es ist deutlich zu sehen, dass die gewöhnlichen Schwerewellen die größte Energie bergen. Deshalb werden im Rahmen dieser Arbeit eben diese Wellen erzeugt. Es sei an dieser Stelle noch angemerkt, dass prinzipiell alle Wellen zur Energieerzeugung geeignet sind. Beispielsweise wird in [Gra95] näher auf die Nutzung von Gezeitenwellen eingegangen, die allerdings oftmals zu große Auswirkungen auf die Natur hat und deshalb nicht immer sinnvoll ist.

Wie schon aus der Abbildung ersichtlich ist, entstehen Schwerewellen durch Wind. Dabei sind drei Faktoren entscheidend für die Form und Höhe der Welle: Windgeschwindigkeit, Winddauer und Wellenlänge. Der über die Wasseroberfläche hinweg wehende Wind schiebt aufgrund der Reibung die obersten Wasserteilchen an. Durch Turbulenzen in der Luft entstehen so Druckunterschiede an der Wasseroberfläche, die sich dadurch hebt und senkt, um diese Unterschiede

auszugleichen. Diese kleinen Unebenheiten in der Wasseroberfläche werden durch den anhaltenden Wind noch mehr verstärkt, bis immer größere, nahezu periodische Wellen entstehen.

Theoretisch könnten so bei konstantem Wind immer größere Wellen entstehen. Tatsächlich ist die Wellenhöhe jedoch durch die Wellenperiode und Wassertiefe begrenzt. Ist nämlich der Punkt erreicht, an dem die Teilchen an der Wasseroberfläche eine höhere Geschwindigkeit haben als die Welle selbst, so überschlägt sich die Welle. Durch die entstehenden Turbulenzen beim Wellenbrechen geht auch ein Teil der Wellenergie verloren und die Wellen können nicht weiter in die Höhe wachsen. Ist so eine konstante Wellenhöhe erreicht, bleibt auch die Wellenperiode konstant und der Wellengang wird als *ausgereifter Seegang* bezeichnet. Die Gesamtenergie der Wellen bleibt nahezu gleich und bewegt sich mit Gruppengeschwindigkeit c_G fort. Der Name Gruppengeschwindigkeit rührt daher, dass Wasserwellen im Allgemeinen nie als einzelne monochrome Wellen auftreten, sondern aus mehreren überlagerten Wellen bestehen. Die Energie bleibt also auch erhalten, wenn der Wind aussetzt. Solche Wellen, die nicht mehr von zeitnahen Einflüssen, also in diesem Fall dem Wind, abhängen, werden *Dünung* genannt.

Eine Änderung der Wellenlänge und -geschwindigkeit findet erst im flacher werdenden Wasser statt. Dort hat die Wassertiefe entscheidenden Einfluss auf die Welle, wie in Abbildung 2.1.2 zu sehen ist. Die Wasserteilchen bewegen sich in tiefem Wasser in einer Kreisbewegung. Betrachtet man also einen Partikel an der Wasseroberfläche, so bewegt sich dieser im Kreis während einer Wellenbewegung. Mit flacher werdendem Wasser verändert sich diese Bewegung zu einer Ellipse, wie auch in der Abbildung erkennbar ist. Die Unterteilung in Flach- und Tiefwasser hängt hierbei von der *relativen Wellenhöhe* $\frac{d}{L}$ ab, wobei d die Wassertiefe und L die Wellenlänge bezeichnet. Diese beeinflusst auch die Gruppengeschwindigkeit c_G , mit der sich die Wellen fortbewegen.

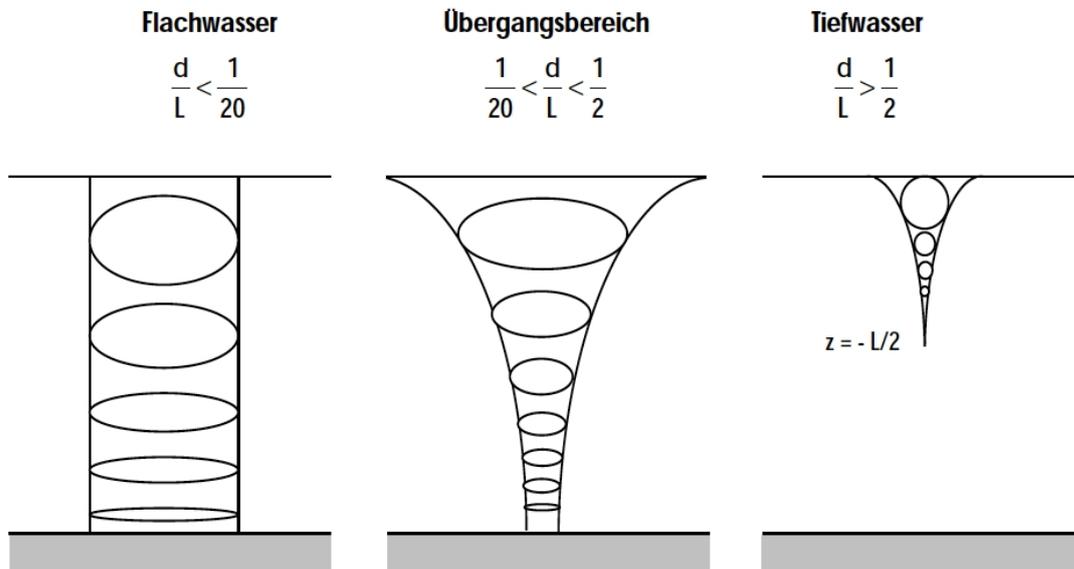


Abb. 2.1.2: Einfluss der Wassertiefe auf die Wellen. Die Einteilung in Flachwasser, Tiefwasser und Übergangsbereich erfolgt über die relative Wassertiefe d/L . Die Ellipsen stellen die Bewegung der Wasserteilchen dar (aus [Gra95]).

Kommt eine Welle in den Flachwasserbereich, das heißt die relative Wellenhöhe ist kleiner als $\frac{1}{20}$, so verliert sie an Länge und Geschwindigkeit. Ein Teil der Energie geht durch Reibung am Boden verloren und ein Teil durch Turbulenzen beim Brechen der Welle, sobald ihre Geschwindigkeit weit genug gesunken ist. Oftmals wird auch die Laufrichtung der Welle verändert, wie zum Beispiel an Stränden zu beobachten ist. Dort kommen die Wellenkämme immer nahezu parallel zum Strand an. Dieses Phänomen lässt sich ebenfalls durch den soeben beschriebenen Einfluss der Wassertiefe auf die Wellenenergie und insbesondere auf die Geschwindigkeit erklären.

2.2 Wellenkraftwerke

Ebenso wie es verschiedene Wellen gibt, gibt es auch eine Reihe verschiedener Wellenkraftwerke. Diese können beispielsweise nach Bauweise (zum Beispiel schwimmend oder fest am Boden montiert), Art der Energieumwandlung (zum Beispiel von Luft oder Wasser angetriebene Turbine oder bewegliche Hydraulikelemente) oder auch nach ihrer Entfernung zur Küste unterteilt werden.

Wie in Abbildung 2.2.1 zu sehen ist, gibt es viele Möglichkeiten, solche Kraftwerke zu bauen. Einige von den hier dargestellten wurden bereits gebaut oder befinden sich derzeit noch in der Testphase.

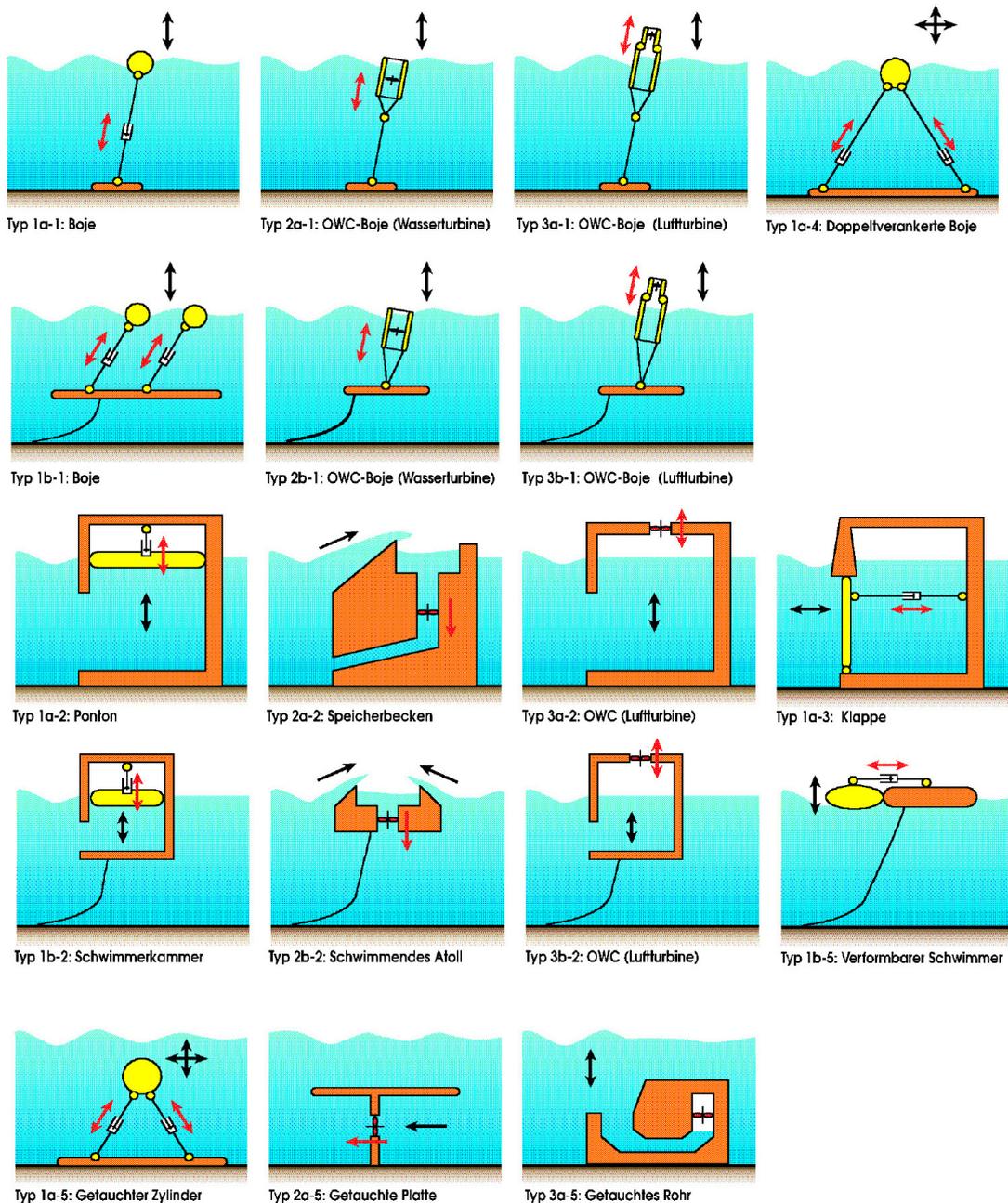


Abb. 2.2.1: Verschiedene Arten von Wellenkraftwerken. Die Einteilung erfolgt über die Bauweise (fest oder schwimmend, getaucht oder nicht getaucht) und die Funktionsweise (nach [Gra95]).

2.2.1 Pelamis

Ein gutes Beispiel für einen schwimmendes Wellenenergiewandler ist die von einer Schottischen Firma gebaute Seeschlange Pelamis, die bereits vor der portugiesischen Küste getestet wurde. Dort wurde sogar eine ganze Farm bestehend aus einem Netz von mehreren dieser Riesenschlangen gebaut. Derzeit laufen mehrere Projekte in verschiedenen Ländern, darunter zwei solcher Farmen vor der Küste Schottlands und auch ein von der Firma E.ON UK in Auftrag gegebenes Projekt zum Testen des neuen P2 Pelamis Generators [Pel]. Die Energie wird hierbei durch Hydrauliksysteme — in Abbildung 2.2.2 schwarz gekennzeichnet — zwischen den einzelnen Segmenten der Schlange gewonnen. Die Elemente bewegen sich durch die Wellenbewegung gegeneinander und treiben auf diese Weise die Energieumwandlung an. Der produzierte Strom gelangt über am Meeresgrund verlegte Kabel an das Festland.

Ein Vorteil dieser Wellenenergiewandler ist die Robustheit gegenüber enorm hohen Wellen. Die Seeschlange arbeitet am besten bei einem moderaten Seegang und ist in der Lage, bei zu großen Wellen unterzutauchen.



Abb. 2.2.2: Pelamis Wellenkraftwerk. Die "Seeschlange" wird durch Hydraulikelemente zwischen den einzelnen Segmenten angetrieben (eigene Abbildung).

2.2.2 Oscillating–Water–Column

Eine andere Art von Energieumwandlung machen sich sogenannte *Oscillating–Water–Column* (OWC) Systeme zu nutze. Dabei ist das Kraftwerk so gebaut, dass ein Teil unter Wasser und ein Teil oberhalb liegt. Der Innenteil besteht aus einem Hohlraum, in dem durch die Lage des Kraftwerks eine Wassersäule entsteht. Diese bewegt sich durch die ankommenden Wellen auf und ab — also oszilliert — und die Luft über dieser Wassersäule wird dadurch ebenfalls in Bewegung versetzt. Durch diesen Luftstrom wird eine im Kraftwerk installierte Turbine angetrieben. Die Idee hierbei ist, dass die große Kraft der Wasserwellen, die mit einer niedrigen Geschwindigkeit auf das Kraftwerk treffen, in eine kleine Kraft mit einer großen Geschwindigkeit umgewandelt wird [Gra95]. Eben diese Bedingungen benötigen die herkömmlichen Generatoren.

Wie in Abbildung 2.2.3 deutlich wird, bewegt sich die Luft bei diesem Vorgang — je nachdem ob sich die Welle hebt oder senkt — in verschiedene Richtungen; deshalb werden spezielle Turbinen eingesetzt, wie beispielsweise die sogenannte *Wells–Turbine*. Diese ist so konzipiert, dass sie sich immer in die gleiche Richtung dreht, gleichgültig aus welcher Richtung der antreibende Luftstrom kommt. Dabei geht zwar auch Energie verloren, aber die Turbine kann sich konstant drehen und Strom produzieren.

Es gibt viele Möglichkeiten solche OWC–Kraftwerke zu bauen, wie auch in Abbildung 2.2.1 zu sehen ist: OWC–Bojen (Typ 2a-1, 2b-1 sowie Typ 3a-1, 3b-1), fest installierte OWC–Bauwerke (Typ 3a-2) oder auch schwimmende OWC–Kraftwerke (Typ 3b-2) sind vorstellbar.

Das erste installierte Kraftwerk dieser Art ist das auf der schottischen Insel Islay gebaute *LIMPET* (*Land Installed Marine Powered Energy Transformer*) der Firma Voith Hydro [Voi] und ist ein fest am Boden installiertes Bauwerk, wie Typ 3a-2 in Abbildung 2.2.1. Seit 2000 produziert es bereits ohne Unterbrechung Energie für das öffentliche Stromnetz in Großbritannien und ist Vorbild für viele weitere Wellenenergiewandler dieser Art.

OWC–Kraftwerke gibt es nicht nur in Küstennähe und sie müssen auch nicht fest am Boden installiert sein. Ein gutes Beispiel dafür sind die bereits erwähnten OWC–Bojen, die völlig autark Strom produzieren und so eine an der Oberseite befestigte Glühbirne betreiben können. Diese Bojen werden bereits seit Jahrzehnten erfolgreich eingesetzt und beispielsweise von der japanischen Firma Ryokuseisha in verschiedenen Varianten vertrieben [Gra06].

Ein wichtiger Faktor hinsichtlich der Energiegewinnung ist die Frequenz der Wellen. Die OWC–Kammer sollte zur optimalen Energieausbeute auf die Resonanz des Seegangs abgestimmt sein. McIver und Evans [ME88] beschreiben beispielsweise einen Ansatz, bei dem Leitwände vor dem Kraftwerk für eine bessere Nutzung der Wellenenergie sorgen sollen. Diese Wände sollen die Wellen

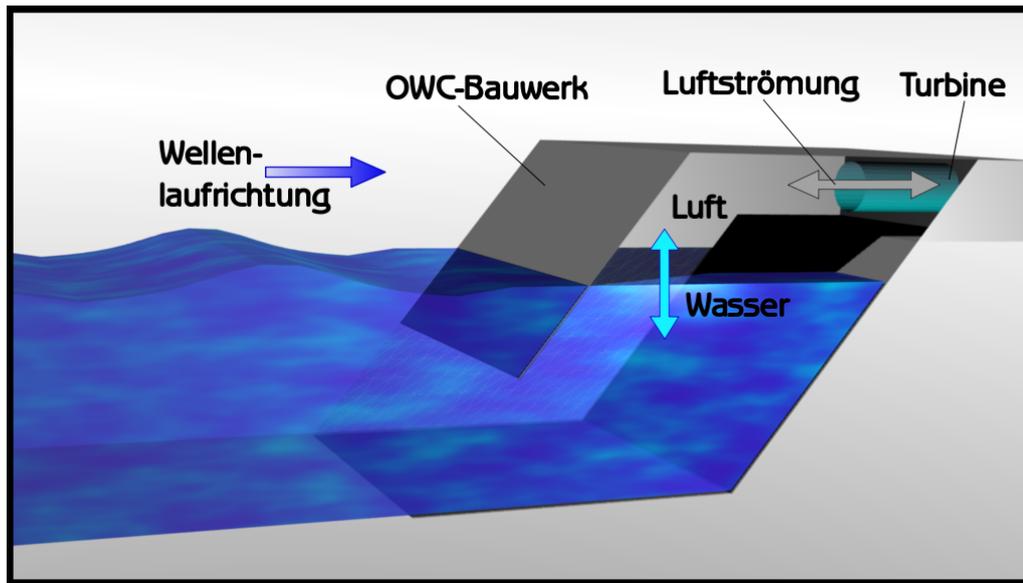


Abb. 2.2.3: Funktionsweise eines *Oscillating-Water-Column* Kraftwerkes. Die bewegte Luftsäule im Inneren des Bauwerks treibt eine Turbine an, die an einen Generator angeschlossen ist. Auf diese Art wird Strom erzeugt (eigene Abbildung).

bündeln und verstärken, um die Energieabsorption zu maximieren. Falnes [Fal02] beschreibt verschiedene Ansätze, um eine Abstimmung des Wellenkraftwerks auf die Wellenfrequenz zu erhalten. Dabei wird die Theorie einerseits für gleichmäßige sinus-förmige Wellen und andererseits für unregelmäßige Wellen, wie sie an der Meeresoberfläche auftreten, formuliert. Außerdem wird zwischen *kontinuierlicher* und *diskreter* Kontrolle unterschieden. Letzteres ermöglicht die Abstimmung auf die Oszillation nur zu bestimmten Zeitpunkten, während die erste Möglichkeit eine fortlaufende Kontrolle bietet.

Eine diskreter Ansatz nutzt beispielsweise das sogenannte *latching*, welches auch in [Gra95] näher beleuchtet wird. Dabei wird zu bestimmten Zeitpunkten die Luftzirkulation unterbrochen, um die Phase und Amplitude des Luftstroms zu regulieren. Um die Abstimmung auf die Resonanz des Seegangs kontinuierlich zu ermöglichen, ist vor allem die Dimension der Kammer des OWC-Kraftwerks entscheidend. Graw stellt außerdem noch einen Druckspeicher im Turbinenzulauf, flexible Festkörper vor sowie in der Kammer oder den Einsatz von zusätzlichen Kammern innerhalb des Kraftwerks vor.

Alle diese Ansätze setzen dabei voraus, dass die Wellenfrequenz zumindest bis zu einem gewissen Punkt in die Zukunft absehbar ist. Vereinfacht kann für unflexible Konstruktionen ohne aktive Kontrolle der Resonanz das Bauwerk auf die durchschnittliche Frequenz der Wellen an dem entsprechenden Aufstellungsort abgestimmt werden. Doch auch diese Vorgehensweise ist äußerst kompliziert, da die Resonanz des Wellenkraftwerks von vielen Faktoren, wie beispielsweise Volumen der Kammer, Volumen des Wassers vor und in der Kammer, Dimension der Turbine und auch Widerstand von Luft und Wasser in bestimmten Bereichen des Wellenkraftwerks abhängt.

Da das Grundprinzip der OWC-Generatoren sehr einfach und in Computersimulationen sehr gut zu testen ist, werden in dieser Arbeit diese Kraftwerke betrachtet. Dazu sei an dieser Stelle bereits angemerkt, dass die Luft im Hohlraum des Bauwerkes je nach eingesetzter Turbine komprimiert und somit der Druck im Inneren erhöht wird. Dies kann unter Umständen auch dazu führen, dass sich die auf die Turbine übertragene Energie erhöht; dies ist aber von der jeweiligen Turbine abhängig. Da in dieser Arbeit die *inkompressiblen* Navier-Stokes-Gleichungen betrachtet werden, wird dieser Teil vernachlässigt und lediglich der entstehende Luftstrom beobachtet. Dies führt aus den beschriebenen Gründen nur bedingt zu einem Modellfehler, da im Rahmen dieser Arbeit keine speziellen Turbinen simuliert werden, sondern eine offene Kammer, aus der die Luft ausströmen kann.

2.3 Numerische Wellenbecken

Wie können nun also Wellen simuliert werden, um die beschriebenen Wellenkraftwerke zu testen? Eine Lösung bieten sogenannte Numerische Wellenbecken (*NWT* — *Numerical Wave Tank*). Dabei wird zwischen zwei Arten von Wellenbecken unterschieden, je nachdem auf welche Art die Wellen erzeugt werden:

- Werden die Wellen durch Veränderungen der zugrunde liegende Gleichungen (zum Beispiel durch eine Modifizierung der Randbedingungen oder durch Setzen einer anderen rechten Seite innerhalb eines kleinen Gebietes) generiert, wird dies als eine *rein numerischen Wellenerzeugung* bezeichnet.
- Werden die Wellen jedoch durch Reproduktion eines physikalischen Wellenbeckens produziert, wird dies *physikalische Wellenerzeugung* genannt. In solchen Becken hebt und senkt sich der Wasserspiegel aufgrund von Wasserverdrängung. Das heißt ein Wellenerzeuger bewegt sich in einem bestimmten Rhythmus und generiert auf diese Weise Wellen.

In dieser Arbeit werden letztere nachgebaut. Ein detaillierter Überblick über die verschiedenen Numerischen Wellenbecken und deren Umsetzung in Strömungssimulationen, sowie die zugrunde liegende Theorie findet sich in [KCT99] und [Tan00].

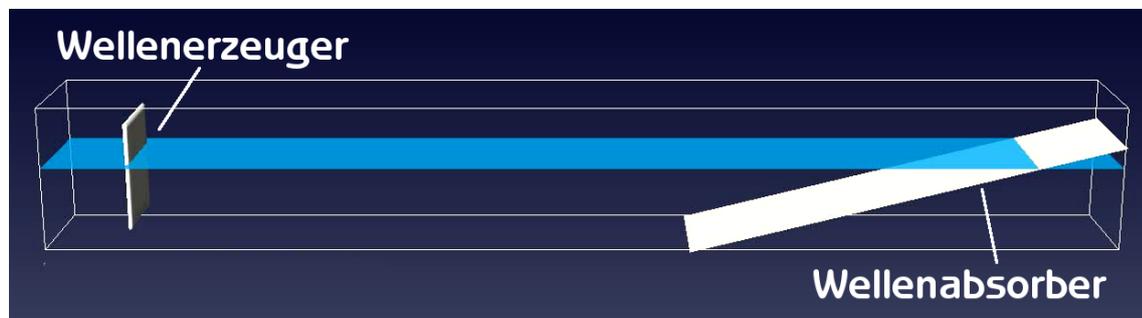


Abb. 2.3.1: Typischer Aufbau eines Numerischen Wellenbeckens. An einem Ende befindet sich ein Wellenerzeuger und am anderen Ende ein Wellenabsorber, der ungewollte Wellenreflexionen reduziert (eigene Abbildung).

Ein Numerisches Wellenbecken hat einen charakteristischen Aufbau, bestehend aus einem Wellenerzeuger (*Wave-Maker*) auf der einen Seite und einem Wellenabsorber (*Wave-Absorber*) auf der anderen Seite, um eine Reflexion am Beckenrand zu vermeiden. Dazwischen muss genügend Abstand liegen, um eine ausgereifte Dünung zu erhalten, da in der Nähe des Wellenerzeugers noch abklingende Wellen zu beobachten sind, die jedoch nach einem Abstand von dreifacher Wassertiefe vernachlässigbar sind (vgl. [DD90]). Außerdem bestimmt die Beckenlänge auch die maximale Zeit bevor die Wellen am Beckenrand reflektiert werden. Ein typischer Aufbau von den hier verwendeten Wellenbecken ist in Abbildung 2.3.1 zu sehen.

2.4 Wellenerzeuger Theorie

Je nach Wellenerzeuger und Wassertiefe werden verschiedene Wellenhöhen erwartet. Eine ausführliche Theorie dazu wird beispielsweise in Dean und Dalrymple [DD90] beschrieben.

Es wird beispielhaft ein Stempel-Wellenerzeuger (*piston-type wavemaker*), wie in Abbildung 2.3.1 dargestellt, mit Hubweg S und Wassertiefe h betrachtet. Dabei wird der Stempel in Wellenrichtung hin und her bewegt. Das Volumen des verdrängten Wassers nach einer vollen Bewegung des Stempels beträgt dann Sh (vgl. Abbildung 2.4.1) und das Volumen eines Wellenberges ist durch

$$\int_0^{L/2} (H/2) \sin(kx) dx = H/k \quad (2.4.1)$$

gegeben, wobei L die Wellenlänge, H die Wellenhöhe und $k = \frac{2\pi}{L}$ die Wellenperiode ist. Daraus ergibt sich das Verhältnis

$$Sh = \frac{H}{k} = \frac{H}{2} \left(\frac{L}{2} \right) \frac{2}{\pi}, \quad (2.4.2)$$

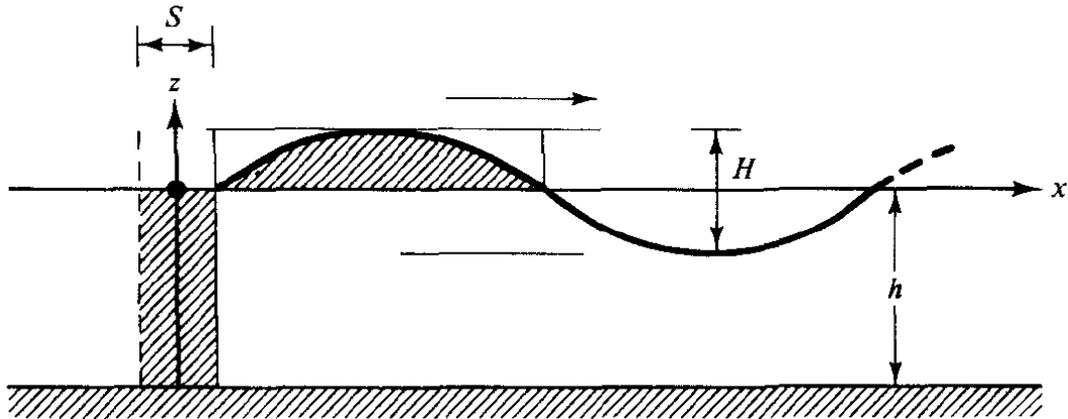


Abb. 2.4.1: Schema eines Stempel-Wellenerzeugers im Flachwasser in der $x - z$ -Ebene. Dabei ist h die Wassertiefe, H die Wellenhöhe und S der Hubweg des Stempels (aus [DD90]).

wobei der Faktor $2/\pi$ das Verhältnis der grau markierten Fläche des Wellenberges zu dem umschließenden Rechteck ist. Aus dieser Gleichung ergibt sich

$$\left(\frac{H}{S}\right)_{piston} = kh, \quad (2.4.3)$$

wobei nun H/S das Verhältnis von Höhe zu Hubweg bezeichnet. Das gilt allerdings nur im Flachwasserbereich, das heißt es muss $\frac{h}{L} < \frac{1}{20}$ beziehungsweise $kh < \frac{\pi}{10}$ gelten. Wird nun einen Klappen-Wellenerzeuger (*flap-type wavemaker*) betrachtet, der am Boden befestigt ist, so ändert sich das Volumen des verdrängten Wassers um den Faktor 2, das heißt es gilt

$$\left(\frac{H}{S}\right)_{flap} = \frac{kh}{2}. \quad (2.4.4)$$

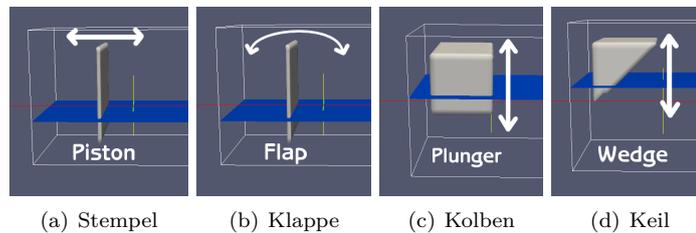


Abb. 2.4.2: Verschiedene Arten von Wellenerzeugern. Diese vier werden in dieser Arbeit zur Wellengenerierung eingesetzt (eigene Abbildung).

In dieser Arbeit werden vier verschiedenen Arten von Wellenerzeugern untersucht: Stempel (*piston*), Klappe (*flap*), Kolben (*plunger*) und Keil (*wedge*), wie in Abbildung 2.4.2 dargestellt.

Ebenso wie für Wellenerzeuger, gibt es auch für Wellenabsorber eine ausführliche Theorie. Es wird hier aber nur eine einfache Schräge betrachtet, die durch Haftrandbedingungen genügend Reibung erzeugt, um die ankommenden Wellen aufzuhalten. Die Wellenreflexion ist dadurch zwar nicht vollständig eliminiert, aber es reicht aus, um regelmäßige Wellen zu produzieren. Diese Art von Wellenabsorber wird in der Literatur oftmals als *Strand* bezeichnet, aufgrund seiner offensichtlichen Ähnlichkeit zu einem natürlichen Ufer.

Kompliziertere numerische Wellenabsorber ahmen entweder das Verhalten des gegenüber liegenden Wellenerzeugers nach oder versuchen durch numerisches Dämpfen — beispielsweise durch ein zum Rand hin größer werdendes Gitter — die Wellen aufzuhalten.

3 Strömungssimulation mit NaSt3DGPF

In diesem Kapitel werden die zugrunde liegenden Gleichungen für die Strömungssimulation, sowie deren Diskretisierung beschrieben. Außerdem wird der verwendete Strömungslöser *NaSt3DGPF* für die Berechnung von inkompressiblen Zweiphasenströmungen mit freier Oberfläche erläutert.

3.1 Die Navier–Stokes–Gleichungen

Die Navier–Stokes–Gleichungen beschreiben das Verhalten instationärer, das heißt zeitabhängiger, Strömungen von viskosen Fluiden. Als Fluid wird ein Gas oder eine Flüssigkeit bezeichnet und die Viskosität gibt an, wie zähflüssig dieses ist. Honig beispielsweise hat eine sehr hohe Viskosität im Gegensatz zu Wasser. Außerdem wird in dem hier betrachteten Modell noch angenommen, dass alle betrachteten Fluide inkompressibel, also nicht zu komprimieren, sind. Für Wasser trifft dies zwar auch annähernd zu, für Gase wie Luft beispielsweise allerdings nicht. Daher ergibt sich ein erster Modellfehler, der bei späteren Auswertungen berücksichtigt werden muss. Eine genaue Herleitung der Navier–Stokes–Gleichungen aus Masse- und Impulserhaltung findet sich beispielsweise in [Cro10].

Es wird zunächst ein einphasiges Modell, das heißt mit nur einem Fluid, betrachtet und es ergibt sich ein System partieller Differentialgleichungen zweiter Ordnung. Dieses besteht aus den *Impulsgleichungen*

$$\rho(\partial_t \mathbf{u} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})) + \nabla p = \nabla \cdot \mu \mathbf{D} + \rho \mathbf{g} \quad (3.1.1a)$$

und der *Kontinuitätsgleichung*

$$\nabla \cdot \mathbf{u} = 0 \quad (3.1.1b)$$

mit Geschwindigkeitsvektor $\mathbf{u} = (u, v, w)^T$, $\mathbf{u} : \Omega \times [0, T] \rightarrow \mathbb{R}^3$, konstanter Dichte ρ , Druck $p : \Omega \times [0, T] \rightarrow \mathbb{R}$, Viskosität μ und Volumenkräften $\mathbf{g} = (g_1, g_2, g_3)^T$. Hierbei ist das Tensorprodukt zweier Vektoren als 3×3 Matrix mit den Komponenten $(a \otimes b)_{i,j} := a_i b_j$ für $i, j = 1, 2, 3$ definiert und $\mathbf{D} = \nabla \mathbf{u} + \{\nabla \mathbf{u}\}^T$ bezeichnet den Deformationstensor.

Das ganze wird in dem Gebiet $\Omega \subset \mathbb{R}^3$ betrachtet. Daher kann das Gleichungssystem (3.1.1) auch folgendermaßen ausgeschrieben werden:

$$\begin{aligned} \partial_t u + \frac{1}{\rho} \partial_x p &= \frac{\mu}{\rho} (\partial_{xx} u + \partial_{yy} u + \partial_{zz} u) - \partial_x (u^2) - \partial_y (uv) - \partial_z (uw) + \mathbf{g}_x \\ \partial_t v + \frac{1}{\rho} \partial_y p &= \frac{\mu}{\rho} (\partial_{xx} v + \partial_{yy} v + \partial_{zz} v) - \partial_x (uv) - \partial_y (v^2) - \partial_z (vw) + \mathbf{g}_y \\ \partial_t w + \frac{1}{\rho} \partial_z p &= \frac{\mu}{\rho} (\partial_{xx} w + \partial_{yy} w + \partial_{zz} w) - \partial_x (uw) - \partial_y (vw) - \partial_z (w^2) + \mathbf{g}_z \\ \partial_x u + \partial_y v + \partial_z w &= 0 \end{aligned}$$

3.2 Zweiphasenströmung

Das Strömungsmodell für zwei Phasen beruht auf den Navier–Stokes–Gleichungen; hinzu kommen aber noch Randbedingungen entlang der Grenzschicht zwischen den beiden Fluiden. So wird auch der Einfluss von Oberflächenspannungskräften berücksichtigt.

Es werden nun also zwei Fluide, die sich nicht vermischen können, betrachtet, wie beispielsweise Luft und Wasser oder auch Öl und Wasser. Dabei entsteht zwischen den beiden Fluiden eine Grenzschicht, die als *freie Oberfläche* bezeichnet wird. Es werden also die inkompressiblen Navier–Stokes–Gleichungen auf der offenen Menge $\Omega := \Omega_1 \cup \Omega_2 \cup \Gamma_f \subset \mathbb{R}^3$ mit Lipschitz–Rand $\Gamma = \partial\Omega$ betrachtet. Dabei bezeichnen Ω_1 und Ω_2 die jeweiligen Teilgebiete der beiden Fluide und $\Gamma_f := \partial\Omega_1 \cap \Omega_2$ die freie Oberfläche, wobei mit dem Rand der jeweiligen Menge der Rand bezüglich des Abschlusses der Menge gemeint ist. Die beiden Teilgebiete sowie der freie Rand sind zeitabhängig.

Hierbei wird das Verhalten der beiden Fluide durch die einphasigen Navier–Stokes–Gleichungen in dem jeweiligen Teilgebiet beschrieben und es ergibt sich das Gleichungssystem

$$\rho_i \frac{D\mathbf{u}_i}{Dt} = -\nabla p_i + \nabla \cdot (\mu_i \mathbf{D}_i) + \rho_i \mathbf{g} \quad \text{in } \Omega_i, \quad (3.2.1a)$$

$$\nabla \cdot \mathbf{u}_i = 0 \quad \text{in } \Omega_i, \quad (3.2.1b)$$

$$\mathbf{u}_i|_{\Gamma} = 0 \quad \text{in } [0, T], \quad (3.2.1c)$$

$$\mathbf{u}_i|_{t=0} = \mathbf{u}_{0_i} \quad \text{in } \Omega_i, \quad (3.2.1d)$$

wobei $i \in \{1, 2\}$ für die jeweilige Fluidphase und $\frac{D\mathbf{u}}{Dt} = \frac{d\mathbf{u}}{dt} + \mathbf{u} \cdot \nabla \mathbf{u}$ die Materialableitung bezeichnet. Dabei wird angenommen, dass die Viskosität μ_i und die Dichte ρ_i jeweils konstant sind. Um die Zweiphasenströmung vollständig zu beschreiben fehlen jedoch noch weitere Bedingungen auf der freien Oberfläche Γ_f . Deshalb wird weiterhin gefordert

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{auf } \Gamma_f, \quad (3.2.2a)$$

$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \mathbf{n} = \sigma \kappa \mathbf{n} \quad \text{auf } \Gamma_f. \quad (3.2.2b)$$

Das Zweiphasensystem (3.2.1) wird also um die *kinematische Bedingung* (3.2.2a) ergänzt, die besagt, dass die freie Oberfläche eine scharfe Grenzschicht bildet und keine Masse hindurch fließen kann, und somit insbesondere die Geschwindigkeiten entlang der Trennschicht stetig sind. Außerdem kommt noch die *dynamische Bedingung* hinzu, die einen Sprung der Dichte und Viskosität entlang des freien Randes beschreibt. In diesem Fall reduziert sich diese auf Gleichung (3.2.2b) mit Spannungstensor $\mathbf{T}_i := -p_i \mathbf{I} + \mu_i \mathbf{D}_i$ (\mathbf{I} ist hier die Einheitsmatrix), Oberflächenspannungskoeffizient σ , der hier als Materialkonstante betrachtet wird, lokaler Krümmung κ und Einheitsnormalenvektor \mathbf{n} . Nähere Ausführungen dazu finden sich in [Cro10].

Gleichungen (3.2.1a) und (3.2.1b) lassen sich zusammen mit Gleichung (3.2.2b) in eine Integralgleichung einbinden und es ergibt sich die zugehörige differentielle Form

$$\rho(\phi) \frac{Du}{Dt} + \nabla p = \nabla \cdot (\mu(\phi) \mathbf{D}) - \sigma \kappa \delta(\phi) \mathbf{n} + \rho(\phi) \mathbf{g}, \quad (3.2.3)$$

wobei ϕ die *Level-Set-Funktion* ist, welche die freie Oberfläche als $\Gamma_f(t) := \{\mathbf{x} \in \Omega | \phi(\mathbf{x}, t) = 0\}$ beschreibt und in Abschnitt 3.4 noch näher erläutert wird, und δ die eindimensionale Dirac δ -Funktion. Da Gleichung (3.2.3) auf dem gesamten Gebiet Ω betrachtet wird, sind Dichte und Viskosität nun nicht mehr konstant, sondern weisen einen Sprung entlang der Grenzschicht zwischen den beiden Phasen auf. Deshalb werden diese beiden Größen in Abhängigkeit von ϕ beschrieben durch

$$\begin{aligned} \rho(\phi) &= \rho_1 + (\rho_2 - \rho_1)H(\phi), \\ \mu(\phi) &= \mu_1 + (\mu_2 - \mu_1)H(\phi). \end{aligned}$$

Hierbei ist $H(\phi)$ die Heaviside Sprungfunktion und ist definiert durch

$$H(\phi) := \begin{cases} 0 & \text{falls } \phi < 0 \\ \frac{1}{2} & \text{falls } \phi = 0 \\ 1 & \text{falls } \phi > 0 \end{cases}. \quad (3.2.5)$$

Die Idee zur Einbindung der Oberflächenspannungskräfte in die Impulsgleichungen mittels der sogenannten *Continuum–Surface–Force (CSF)*-Methode wurde zum ersten mal von Brackbill et al. [BKZ92] für *Volume-of-Fluid (VOF)*-Methoden vorgeschlagen. Eine Darstellung wie in Gleichung (3.2.3) findet sich auch in [Cro10], [SSO94] oder [CHMO96].

3.3 Randbedingungen

Um ein wohlgestelltes Anfangs–Randwertproblem zu erhalten, müssen noch Randbedingungen für das Geschwindigkeitsfeld \mathbf{u} auf dem Gebietsrand $\partial\Omega$ gestellt werden. Dabei wird im Wesentlichen zwischen drei Arten von Randbedingungen unterschieden. Bei *Dirichlet–Randbedingungen* wird

die Variable am Rand auf einen bestimmten Wert gesetzt und bei *Neumann-Randbedingungen* wird der Gradient in eine Richtung vorgegeben. Außerdem sind auch Kombinationen aus beiden Konditionen möglich, die als *Robin-Randbedingungen* bezeichnet werden. Durch verschiedene Vorgaben für die Randwerte ändert sich die Lösung des Gleichungssystems und es ergibt sich je nach Randbedingungen eine andere Strömung. Dabei kann auf jedem Randabschnitt eine andere Randbedingung gesetzt werden. Auf diese Weise können folgende physikalische Eigenschaften auf den Rändern simuliert werden:

- **Haftbedingung** (*no-slip*): Das Fluid bleibt am Rand haften, das heißt der Geschwindigkeitsvektor verschwindet:

$$\mathbf{u}|_{\partial\Omega} = 0.$$

- **Ein- oder Ausströmbedingung** (*in-/outflow*): Das Fluid strömt mit einer vorgegebenen Geschwindigkeit u_0 an dem Rand $\Gamma_{in/out} \subset \partial\Omega$ ein oder aus:

$$\mathbf{u}|_{\Gamma_{in/out}} = u_0 \cdot \mathbf{n}.$$

- **Rutschbedingung** (*free-slip*): Das Fluid kann ohne Reibungsverlust am Rand $\Gamma_R \subset \partial\Omega$ entlangströmen, das heißt nur die tangentialen Geschwindigkeiten ohne Änderung in Normalenrichtung bleiben erhalten:

$$(\mathbf{u} \cdot \mathbf{n})|_{\Gamma_R} = 0, \quad \partial_n(\mathbf{u} \cdot \mathbf{t})|_{\Gamma_R} = 0, \quad \partial_n(\mathbf{u} \cdot \mathbf{s})|_{\Gamma_R} = 0.$$

mit Einheitsnormalektor \mathbf{n} und linear unabhängigen Tangentialvektoren \mathbf{t} und \mathbf{s} .

- **Ausströmbedingung** (*outflow*): Das Fluid strömt an dem Rand $\Gamma_{out} \subset \partial\Omega$ aus, das heißt die Geschwindigkeit in Richtung des äußeren Normalenvektors \mathbf{n} ändert sich nicht:

$$\partial_n \mathbf{u}|_{\Gamma_{out}} = 0.$$

Zu Beachten ist hierbei, dass die *Kompatibilitätsbedingung*

$$0 = \int_{\Omega} \nabla \cdot \mathbf{u} \, dx = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, dF \quad (3.3.1)$$

erfüllt ist. Diese ergibt sich durch Integration der Kontinuitätsgleichung (3.1.1b) und Anwendung des Integralsatzes von Gauß. Wird diese Bedingung verletzt, ist die Inkompressibilität des Fluids nicht garantiert und dies kann zu Masseverlusten und somit insgesamt zu Instabilitäten führen.

3.4 Die Level-Set Methode

Wie in Abschnitt 3.2 bereits angedeutet, benötigt die Grenzschicht zwischen zwei Fluiden bei Zweiphasenströmungssimulationen besondere Aufmerksamkeit. Da hier Sprünge in Dichte und Viskosität auftauchen, kann dies zu Instabilitäten führen. Um diese freien Oberflächen möglichst genau und effizient zu Berechnen gibt es verschiedene Methoden. Zwei Beispiele dafür sind die *Marker and Cell (MAC)* Methoden [HW⁺65] oder das *Volume-of-Fluid (VOF)* Verfahren [HN81], die zwei sehr unterschiedliche Ansätze verwenden.

Die im Jahr 1988 von Osher und Sethian [OS88] vorgestellte Level-Set Methode wird in vielen Gebieten, wie Bildverarbeitung, Materialkunde, Robotik und eben auch in der Strömungsdynamik verwendet. Dabei wird versucht, die freie Oberfläche durch eine möglichst glatte skalare Funktion ϕ zu approximieren, deren Nullniveaumenge die gesuchte Oberfläche ist. Somit ist die Darstellung des freien Randes implizit gegeben und dessen Bewegung reduziert sich auf den Transport der skalaren Funktion. Sei nun also $\Omega \subset \mathbb{R}^3$ das gesamte Gebiet und $\Gamma_f \subset \Omega$ eine Fläche, die sich mit fortschreitender Zeit ändert. Diese Fläche soll in Normalenrichtung mittels einer Geschwindigkeitsfunktion transportiert werden. Mit der Idee der Level-Set Methode wird also eine möglichst glatte Funktion $\phi(\mathbf{x}, t)$ gesucht, so dass

$$\Gamma_f(t) := \{\mathbf{x} \in \Omega | \phi(\mathbf{x}, t) = 0\} \quad (3.4.1)$$

für $t \in [t_0, t_{end}]$ gilt. Das heißt, dass alle $\mathbf{x}(t)$ auf dem freien Rand $\Gamma_f(t)$ verschwinden und somit

$$\phi(\mathbf{x}(t), t) = 0 \quad (3.4.2)$$

gilt. Ableiten von Gleichung (3.4.2) mit Hilfe der Kettenregel liefert die Transportgleichung

$$\phi_t + \nabla\phi(\mathbf{x}(t), t) \cdot \mathbf{x}'(t) = 0. \quad (3.4.3)$$

Da in diesem Fall $\mathbf{x}'(t)$ das Geschwindigkeitsfeld \mathbf{u} auf dem freien Rand Γ_f ist, ergibt sich als Anfangswertproblem für den Transport der Level-Set Funktion

$$\phi_t + \mathbf{u} \cdot \nabla\phi = 0, \quad (3.4.4a)$$

$$\phi(\mathbf{x}, 0) = \phi_0. \quad (3.4.4b)$$

Bis jetzt wurde lediglich die Nullniveaumenge der Level-Set Funktion betrachtet. Da eine Level-Set Funktion aber eine vorzeichenbehaftete Abstandsfunktion sein soll, wird zusätzlich noch gefordert, dass

$$\phi(\mathbf{x}, t) = \begin{cases} > 0 & \text{falls } \mathbf{x} \in \Omega_1 \\ = 0 & \text{falls } \mathbf{x} \in \Gamma_f \\ < 0 & \text{falls } \mathbf{x} \in \Omega_2 \end{cases} \quad (3.4.5a)$$

$$\text{und } |\nabla\phi| = 1. \quad (3.4.5b)$$

Gleichung (3.4.5b) wird *Eikonalgleichung* genannt und garantiert, dass ϕ eine Abstandsfunktion ist. Außerdem ist ϕ durch Gleichungen (3.4.5) eine vorzeichenbehaftete Funktion und im gesamten Gebiet Ω Lipschitz-stetig, auch wenn sich Ränder schneiden oder berühren. Das ist ein wichtiger Vorteil der Level-Set Methode, da so durch topologische Veränderungen der freien Oberfläche, also beispielsweise Abtrennen oder Zusammenstoßen von einzelnen Teilen, keine Instabilitäten entstehen, da die Funktion immer noch Lipschitz-stetig ist. Dadurch sind numerische Stabilität und Genauigkeit sicher gestellt.

Ein weiterer Vorteil der Level-Set Methode ist, dass sich Einheitsnormalenvektor und Krümmung des freien Randes einfach über den Gradienten beziehungsweise die Divergenz berechnen lassen durch

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|},$$

$$\kappa = \nabla \cdot \mathbf{n}.$$

Ein Problem ist allerdings, dass die Level-Set Funktion nach dem Transport im Allgemeinen keine Abstandsfunktion mehr ist, das heißt Gleichung (3.4.5b) ist nicht mehr unbedingt erfüllt. Da diese Eigenschaft aber grundlegend ist für die genaue Umsetzung der Oberflächenspannung, muss sie wenigstens in einer ϵ -Umgebung des freien Randes erfüllt sein. Sonst können sehr große oder auch sehr kleine Gradienten der Level-Set Funktion auftreten, was zur Verzerrung der Oberflächenspannung und auch zu nicht physikalischen Oszillationen führen kann.

Dieses Problem kann durch eine Reinitialisierung der Level-Set Funktion behoben werden. Die Idee ist dabei, die Abstandseigenschaft ohne eine Änderung der Nullniveaumenge wieder herzustellen. Diese Vorgehensweise wurde erstmals von Sussman et al. [SSO94] vorgestellt. Dabei wird das Anfangswertproblem

$$d_\tau + \text{sign}(\phi)(|\nabla d| - 1) = 0, \quad (3.4.7a)$$

$$d(\mathbf{x}, 0) = \phi(\mathbf{x}), \quad (3.4.7b)$$

gelöst, mit künstlicher Zeit τ und signum-Funktion

$$\text{sign}(x) = \begin{cases} 1 & \text{falls } x > 0 \\ 0 & \text{falls } x = 0 \\ -1 & \text{falls } x < 0 \end{cases}.$$

Der Vorteil dabei ist, dass die Oberfläche nicht explizit rekonstruiert werden muss und die Nullniveaumenge bleibt erhalten, da $\text{sign}(0) = 0$. Wird ϕ nach dem Transport durch die stationäre Lösung d von Gleichung (3.4.7) ersetzt, bleiben alle geforderten Eigenschaften der Level-Set Funktion erhalten.

Da die Abstandseigenschaft, wie bereits erwähnt, nur in einer ϵ -Umgebung der freien Oberfläche benötigt wird, kann die Zahl der benötigten Iterationen der Reinitialisierung erheblich verringert werden, indem abgebrochen wird, sobald $|\nabla d| = 1$ für $|d| \leq \epsilon$ gilt.

Nach Transport und Reinitialisierung der Level-Set Funktion kann es zu geringen Masseverlusten kommen, die in einigen Fällen durch eine Volumenkorrektur behoben werden können. Näheres dazu wird in Abschnitt 3.5 beschrieben wird.

3.5 Diskretisierung des Zweiphasenmodells

In diesem Abschnitt wird die Diskretisierung der inkompressiblen Navier-Stokes-Gleichungen für zwei Phasen beschrieben. Abschließend wird der vollständige verwendete Algorithmus angegeben.

3.5.1 Das Gitter

Das zu diskretisierende Gebiet Ω ist quaderförmig und kann daher in alle drei Raumrichtungen durch ein äquidistantes Gitter diskretisiert werden. Dabei wird aus Stabilitätsgründen ein *versetztes Gitter* (*staggered grid*) verwendet, das heißt die zu diskretisierenden Größen liegen verschoben zueinander auf dem Gitter. Die skalaren Größen, in diesem Fall Druck und Level-Set Funktion, liegen im Zentrum der Gitterzellen, während die Vektorkomponenten, also u, v und w des Geschwindigkeitsfeldes, auf den Mittelpunkten der Zellseitenflächen liegen, wie in Abbildung 3.5.1 zu sehen ist.

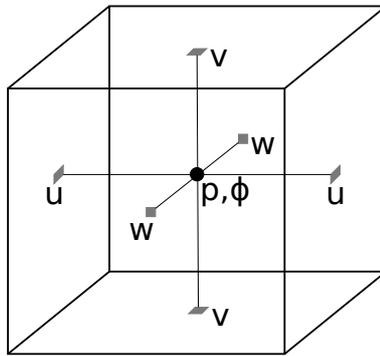


Abb. 3.5.1: Eine Gitterzelle im versetzten Gitter. Die skalaren Größen liegen in den Mittelpunkten der Seitenflächen und die Vektoren im Mittelpunkt der Zelle (eigene Abbildung).

Zu beachten ist außerdem, dass für alle Größen auf dem versetzten Gitter eine zusätzliche Randschicht einführt werden muss, die sogenannten *Geister Punkte* (*Ghost Cells*). Diese gehören nicht zum Gebiet Ω und enthalten die entsprechenden Werte, um die vorgegebenen Randbedingungen zu implementieren.

Die Zeitschrittweite δt wird adaptiv durch sogenannte *Courant-Friedrichs-Lewy-Bedingungen* (*CFL-Bedingungen*) für die verschiedenen Kräfte gesteuert.

3.5.2 Die Level-Set Funktion

Im Folgenden wird die vorzeichenbehaftete Abstandsfunktion ϕ_h betrachtet, die alle Eigenschaften der Level-Set Funktion, also das Gleichungssystem (3.4.5), erfüllt. Da die Dichte und Viskosität an der Phasengrenze unstetig sind, müssen diese zumindest entlang der Sprungschicht Γ_f geglättet werden. Dazu wird die Grenzflächenbreite $\epsilon(\mathbf{x})$ eingeführt, die ungefähr wie die Gittermaschenweite skaliert ist, und die globale Konvergenz erhält. Nur lokal verschlechtert sich die Konvergenzrate um Eins (siehe [Cro10]).

Die beiden kritischen Größen Dichte und Viskosität werden nun also in der $\epsilon(\mathbf{x})$ -Umgebung von Γ_f geglättet und sind in den jeweiligen Teilgebieten außerhalb dieser Umgebung konstant. Somit wird die diskrete Dichte und Viskosität entsprechend ersetzt durch

$$\rho^\epsilon(\phi_h) = \rho_2 + (\rho_1 - \rho_2)H^\epsilon(\phi_h), \quad (3.5.1a)$$

$$\mu^\epsilon(\phi_h) = \mu_2 + (\mu_1 - \mu_2)H^\epsilon(\phi_h), \quad (3.5.1b)$$

wobei $H^\epsilon(\phi_h)$ die geglättete Heaviside Sprungfunktion ist. Diese ist definiert durch

$$H^\epsilon(\phi_h(\mathbf{x})) := \begin{cases} 0 & \text{falls } \phi_h(\mathbf{x}) < -\epsilon(\mathbf{x}) \\ \frac{1}{2}(1 + \frac{\phi_h(\mathbf{x})}{\epsilon(\mathbf{x})} + \frac{1}{\pi} \sin(\frac{\pi\phi_h(\mathbf{x})}{\epsilon(\mathbf{x})})) & \text{falls } |\phi_h(\mathbf{x})| \leq \epsilon(\mathbf{x}) \\ 1 & \text{falls } \phi_h(\mathbf{x}) > \epsilon(\mathbf{x}) \end{cases}. \quad (3.5.2)$$

Diese geglättete Funktion H^ϵ ist nun auch im klassischen Sinne differenzierbar und es ergibt sich als zugehöriges Deltafunktional δ^ϵ die Ableitung der geglätteten Heaviside Funktion,

$$\delta^\epsilon(\phi_h(\mathbf{x})) = \partial_{\phi_h} H^\epsilon(\phi_h(\mathbf{x})) = \begin{cases} \frac{1}{2\epsilon(\mathbf{x})}(1 + \cos(\frac{\pi\phi_h(\mathbf{x})}{\epsilon(\mathbf{x})})) & \text{falls } |\phi_h(\mathbf{x})| \leq \epsilon(\mathbf{x}) \\ 0 & \text{sonst.} \end{cases} \quad (3.5.3)$$

Der Parameter $\epsilon(\mathbf{x})$ wird in Abhängigkeit von Raugittermaschenweite h und Level-Set Funktion ϕ_h gewählt, sodass die globale Konvergenzordnung sich nicht ändert. Details dazu finden sich in [Cro10].

Außerdem werden Oberflächenspannung σ , Krümmung κ und Einheitsnormale \mathbf{n} entsprechend diskretisiert, das heißt es wird $\sigma\kappa\delta(\phi)\mathbf{n}$ in Gleichung (3.2.3) durch $\sigma\kappa_h\delta^\epsilon(\phi_h)\mathbf{n}_h$ ersetzt, mit diskreter Krümmung

$$\kappa_h := \nabla \cdot \mathbf{n}_h$$

und diskreter Einheitsnormale

$$\mathbf{n}_h := \frac{\nabla\phi_h}{|\nabla\phi_h|} = \frac{(\phi_x, \phi_y, \phi_z)_h^T}{\sqrt{(\phi_x^2 + \phi_y^2 + \phi_z^2)_h}}.$$

Pro Zeitschritt wird nun also die Level-Set Funktion über die Transportgleichung

$$\partial_t\phi_h + \mathbf{u}_h \cdot \nabla\phi_h = 0 \quad (3.5.4)$$

transportiert und anschließend werden Dichte und Viskosität aus Gleichung (3.5.1) berechnet.

Die Transportgleichung (3.5.4) wird schließlich durch ein *WENO*-Schema 5. Ordnung für den konvektiven Teil $\mathbf{u} \cdot \nabla\phi_h$ diskretisiert. Für die Zeitdiskretisierung wird ein *Adams-Bashforth* Verfahren 2. Ordnung verwendet. Für nähere Ausführungen dazu sei wieder auf [Cro10] verwiesen.

3.5.3 Reinitialisierung und Masseerhaltung

Um die Reinitialisierung (3.4.7) zu diskretisieren, wird ein *Runge-Kutta* Verfahren 3. Ordnung für die Zeitdiskretisierung sowie ein *WENO* Schema 5. Ordnung im Raum verwendet, welches zu einer besseren Masseerhaltung führt als Verfahren niedrigerer Ordnung [Cro10].

Durch die verwendeten Diskretisierungsverfahren sind bereits Verbesserungen in Hinsicht auf die Masseerhaltung zu beobachten [Cro10]. Da in jedem Zeitschritt eine Reinitialisierung durchgeführt wird, kann es in jedem Zeitschritt zu einer ungewollten Verschiebung der Level-Set Funktion kommen, die in der Größenordnung einer Gittermaschenweite, also in $\mathcal{O}(h)$ liegt. Ein Lösungsansatz dazu ist die *Picard-Fixpunktiteration*, mit der eine Masseerhaltung bis aus Maschinengenauigkeit möglich ist [CGS09].

Um den eventuellen Volumenverlust auszugleichen, wird eine neue Isolinie zum Level-Set Wert $0+c$ betrachtet. Der Wert $|c|$ beschreibt nun also den Abstand der neuen Konturlinie zur alten Null-Konturlinie. Diese Konstante c soll nun so bestimmt werden, dass das Anfangsvolumen $V_1(\phi^0)$ einer Phase Ω_1^0 dem neuen Volumen $V_1(\phi^{n+1}) := \int_{\Omega} H(\phi^{n+1}) dx$ von Ω_1^{n+1} nach der Reinitialisierung entspricht. Die Picard-Fixpunktiteration hat somit die Form

$$\phi^{n+1} \leftarrow \phi^{n+1} + \omega(V_1(\phi^0) - V_1(\phi^{n+1})), \quad (3.5.5)$$

wobei die Differenz $\omega(V_1(\phi^0) - V_1(\phi^{n+1})) := c^{n+1}$ der neuen Wert von c ist, der damit also in jedem Iterationsschritt kleiner wird. Auf diese Weise kann das sich neue Volumen dem Ursprungsvolumen bis auf eine beliebige Genauigkeit annähern.

3.5.4 Die Chorin–Temam–Projektionsmethode

Das Ziel dieses Abschnittes ist es, ein Verfahren vorzustellen, mit dem sich die zweiphasigen Navier–Stokes–Gleichungen (3.1.1) diskretisieren lassen. Das Problem ist dabei, dass sich die Impulsgleichungen (3.1.1a) mit der Kontinuitätsgleichung (3.1.1b) nicht koppeln lassen. Daher wird ein entkoppeltes System hergeleitet, welches es erlaubt Druck und Geschwindigkeit getrennt voneinander zu berechnen. Die Lösung dazu bietet die sogenannte *Chorin–Temam–Projektionsmethode*, die im folgenden vorgestellt wird. Die Idee für die hier verwendete Projektionsmethode wurde von Chorin [Cho68] und unabhängig davon auch von Temam [Té69] entwickelt. Da die Methode ursprünglich für einphasige Systeme entwickelt wurde, muss noch die Oberflächenspannung sowie der Dichte- und Viskositätsprung eingebunden werden. Letztere lassen sich über die Level–Set Funktion berechnen.

Die Grundidee des Verfahrens ist es, die Impulsgleichung (3.1.1a) in einen divergenzfreien und einen rotationsfreien Teil zu zerlegen. Dazu wird die divergenzfreie Schätzgeschwindigkeit u^* mit Hilfe des Drucks in den Raum der divergenzfreien Geschwindigkeiten projiziert.

Im Folgenden wird der Ablauf der Projektionsmethode beschrieben. Zunächst wird das Hilfsgeschwindigkeitsfeld u^* durch

$$u^* = u^n + \delta t(\mathbf{g} - \nabla \cdot (u^n \otimes u^n) + \frac{1}{\rho^\epsilon(\phi^n)}(\nabla \cdot (\mu^\epsilon(\phi^n)\mathbf{D}^n) - \sigma\kappa(\phi^n)\delta^\epsilon(\phi^n)\nabla\phi^n)) \quad (3.5.6)$$

berechnet und anschließend die Level–Set Funktion transportiert, reinitialisiert und eventuell mittels Picard–Fixpunktiteration korrigiert. Dadurch ergibt sich die neue Level–Set Funktion ϕ^{n+1} und das neue Gleichungssystem

$$\frac{u^{n+1} - u^*}{\delta t} + \frac{\nabla p^{n+1}}{\rho(\phi^{n+1})} = 0, \quad (3.5.7a)$$

$$\nabla \cdot u^{n+1} = 0. \quad (3.5.7b)$$

Durch Anwendung des Divergenzoperator auf Gleichung (3.5.7a) und Verwendung von Gleichung (3.5.7b), ergibt sich die *Druck–Poisson–Gleichung*

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^\epsilon(\phi^{n+1})} \right) = \nabla \cdot \frac{u^*}{\delta t}. \quad (3.5.8)$$

Lösen der Poisson–Gleichung (3.5.8) liefert den gesuchten Druck p^{n+1} und u^{n+1} wird durch die Geschwindigkeitskorrektur

$$u^{n+1} = u^* - \frac{\delta t}{\rho^\epsilon(\phi^{n+1})}\nabla p^{n+1} \quad (3.5.9)$$

berechnet.

Damit Gleichung (3.5.8) ein wohlgestelltes Problem ist, fehlt noch die Randbedingung

$$\partial_n p^{n+1}|_{\Gamma_f} = \frac{\rho^\epsilon(\phi^{n+1})}{\delta t}(\mathbf{u}_\Gamma^* - \mathbf{u}_\Gamma^{n+1}) \cdot \mathbf{n}. \quad (3.5.10)$$

Das heißt Gleichung (3.5.7a) wird auf die äußere Einheitsnormale von Ω projiziert. Gilt $u_\Gamma^* = u_\Gamma^{n+1}$, so folgt $\partial_n p^{n+1}|_{\Gamma_f} = 0$ und es ergibt sich eine homogene Neumann–Randbedingung.

Die Poisson–Gleichung (3.5.8) kann mit Hilfe von finiten Differenzen diskretisiert werden und ergibt ein lineares Gleichungssystem der Form $Ax = b$, welches mit verschiedenen Iterationsverfahren gelöst werden kann. Dabei hängt die Kondition des Verfahrens von der Matrix A ab und verschlechtert sich bei größeren Dichtesprüngen an der Phasengrenze.

3.6 Parallelisierung

Durch den Einsatz von Parallelrechnern können heutzutage auch große Simulationen in relativ kurzer Zeit berechnet werden. Dabei wird die Rechnung auf mehrere Prozessoren aufgeteilt, die untereinander kommunizieren können, um nötige Werte auszutauschen. Um einen sequentiellen Code zu parallelisieren, muss daher das betrachtete Problem zunächst aufgeteilt werden, bevor es auf die verschiedenen Prozessoren verteilt werden kann. Da hier ein äquidistantes Gitter in alle drei Raumrichtungen verwendet wird, liegt eine Gebietszerlegung nahe. Das gesamte diskrete

Rechengebiet Ω_h wird demnach auf die zur Verfügung stehenden $1, \dots, P$ Prozessoren aufgeteilt und es ergeben sich die Teilgebiete $\Omega_h^1, \dots, \Omega_h^P$. Dabei sollen sich die Teilgebiete nicht überlappen. Zusätzlich werden diese um eine Randschicht, die sogenannte *Randbordüre* erweitert, um Werte zu speichern, die für die Diskretisierungsverfahren benötigt werden.

Um die Werte in den Randbordüren auf dem aktuellen Stand zu halten, findet in jedem Zeitschritt ein Austausch zwischen den einzelnen Prozessoren statt. Dabei sollte darauf geachtet werden, dass der Aufwand für den Datenaustausch den Berechnungsaufwand nicht übersteigt, da sonst die Parallelisierung ihren Zweck verliert. Aus diesem Grund sollten zum einen die Randbordüren möglichst klein sein. Zum anderen sollten die einzelnen Teilgebiete in etwa gleich groß sein, um den Rechenaufwand möglichst gleichmäßig zu verteilen.

Der hier verwendete Algorithmus wird mit Hilfe der Kommunikationsbibliothek *MPI (Message Passing Interface)* parallelisiert [MPI].

3.7 Der Algorithmus zum Lösen des Zweiphasensystems

In Algorithmus 1 wird der vollständige Algorithmus zusammengefasst dargestellt. Im wesentlichen wird die Chorin–Temam–Projektionsmethode verwendet, die Level–Set Funktion transportiert und reinitialisiert und anschließend eine Picard–Fixpunktiteration zur Volumenkorrektur durchgeführt.

Der gesamte Algorithmus wird wie in Abschnitt 3.6 beschrieben mit MPI parallelisiert.

Algorithmus 1 Paralleler Löser für inkompressible Zweiphasensysteme (nach [Cro10])

Require: Gittermaschenweiten $\delta x, \delta y, \delta z \in \mathbb{R}$, Endzeit T , Level–Set Funktion ϕ_0 , Gebiet Ω

- 1: Setze die lokalen Teilgebiete $\Omega^q, q \in \{1, \dots, P\}$.
 - 2: Setze $t := 0, n := 0$ und $h := \max(\delta x, \delta y, \delta z)$.
 - 3: Initialisiere $u^n := u_0, p^n := p_0$ und $\phi^n := \phi_0$ auf allen Teilgebieten $\Omega^q, q \in \{1, \dots, P\}$.
 - 4: Berechne das Volumen von Ω_1^0 .
 - 5: Setze Grenzschichtweite $\epsilon(\mathbf{x})$.
 - 6: Setze Randwerte für u^n auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 7: **Zeitschleife:**
 - 8: **while** $t \leq T$ **do**
 - 9: **Berechne Zeitschrittweite.**
 - 10: **Impulsgleichung:**
 Kommuniziere Randwerte für u^n und ϕ^n .
 Berechne Hilfgeschwindigkeitsfeld $u^*(u^n, \phi^n)$ nach Gleichung (3.5.6) auf Ω^q .
 Setze Randwerte für u^* auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 11: **Setze rechte Seite der Druck–Poisson–Gleichung:**
 Kommuniziere Randwerte für u^* .
 Setze rechte Seite $\mathbf{b} := \nabla \cdot u^*$ von Gleichung (3.5.8) auf allen Teilgebieten.
 - 12: **Transportgleichung:**
 Transportiere die Level–Set Funktion nach Gleichung (3.5.4).
 - 13: **Reinitialisierung:**
 Führe Reinitialisierung gemäß Gleichung (3.4.7) durch.
 - 14: **Picard–Fixpunktiteration (optional):**
 - 15: **while** $|V_1(\phi^0) - V_1(\phi^{n+1})| \leq tol$ **do**
 - 16: Berechne Volumen von Ω_1^{n+1} .
 - 17: Korrigiere Level–Set Funktion nach Gleichung (3.5.5).
 - 18: **end while**
 - 19: **Druck–Poisson–Gleichung:**
 Berechne p^{n+1} .
 Kommuniziere Randwerte für p_{it} pro Iterationsschritt it .
 - 20: **Geschwindigkeitskorrektur:**
 Berechne u^{n+1} nach Gleichung (3.5.9) auf allen Teilgebieten.
 Setze Randwerte für u^{n+1} auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 21: Setze $t := t + \delta t$ und $n := n + 1$.
 - 22: **end while**
-

4 Numerische Wellenerzeugung

Ein Ziel dieser Arbeit ist es, realistische Wasserwellen zu erzeugen. Daher wird in diesem Kapitel darauf eingegangen, wie dies am Rechner umsetzbar ist. Dazu werden sogenannte *Numerische Wellenbecken*, wie sie bereits in Abschnitt 2.3 erwähnt wurden, eingesetzt. Die verschiedenen Ansätze zur Wellenerzeugung werden kurz erklärt und verglichen, um ein passendes Modell für die Wellengenerierung zu finden. Anschließend wird die Darstellung von Festkörpern und deren Einbindung in das Zweiphasensystem erläutert. Damit kann schließlich ein vollständiger erweiterter Algorithmus angegeben werden, der die Simulation von Wellen ermöglicht.

4.1 Wellenerzeugung am Computer

An dieser Stelle sollen die verschiedenen Methoden zur Wellengenerierung sowie ihre Vor- und Nachteile erläutert werden.

Ein Ansatz von Lin & Liu [LL99] besteht darin, die Kontinuitätsgleichung $\nabla \cdot \mathbf{u} = 0$ lokal zu verändern, sodass auf der rechten Seite eine von Raum und Zeit abhängige Funktion steht. Durch verschiedene Funktionen konnten sie so eine Reihe unterschiedlicher Wellen erzeugen, allerdings lediglich in zwei Dimensionen. Weitere Simulationen in zwei Raumdimensionen wurden beispielsweise von Chen et al. [CKZL99], Wang et al. [WYT95] — dieses Numerische Wellenbecken ist auch als *LONGTANK* bekannt — oder Senturk [Sen11] durchgeführt.

Berechnungen in drei Dimensionen wurden beispielsweise von Kleefsman et al. [KFV⁺05] und Park et al. [PYM⁺01] durchgeführt. Letztere haben dazu ein am Rand vorgeschriebenes Geschwindigkeitsfeld genutzt, um die Bewegung eines Klappen-Wellenerzeugers nachzuahmen. Kleefsman et al. hingegen versuchten es direkt mit der Hilfe von Festkörpern und ließen einen Keil, einen Kolben und eine Kugel ins Wasser fallen, sowie eine fallende Wassersäule, die im Weiteren noch kurz erklärt wird. Dieser Ansatz brachte allerdings nur einzelne Wellen zu Stande.

Es gibt noch eine Reihe weiterer Experimente, um Numerische Wellenbecken sinnvoll zu modellieren. Im weiteren wird darauf eingegangen, wie im Rahmen dieser Arbeit Wellen erzeugt werden und weshalb dieser Ansatz sinnvoll ist.

In Abschnitt 2.1 wurde bereits die Wellenentstehung durch Wind beschrieben. Ein naheliegender Ansatz wäre es daher, eine Strömungssimulation zu berechnen, bei der Wind über eine Wasseroberfläche bläst. Dazu wurden eine Reihe von Versuchen durchgeführt. Zunächst wurde eine moderate Windgeschwindigkeit gewählt, bei der allerdings nur sehr geringe Auslenkungen der Wasseroberfläche zu beobachten waren. Bei höheren Windgeschwindigkeiten brach die Simulation allerdings ab und lieferte keine realistischen Ergebnisse. Abbildung 4.1.1 zeigt eine solche Simulation, bei der die Wasseroberfläche aufgrund der zu hohen Windgeschwindigkeit unregelmäßige Ablösungen aufweist anstatt einen Wellengang zu erzeugen. Diese Art der Wellengenerierung erwies sich als äußerst schwierig. Zum einen müsste der Einflussbereich des Windes so groß gewählt werden, dass das Rechengebiet und somit der Simulationsaufwand sehr groß wird, da wie in Abschnitt 2.1 beschrieben die Wellenhöhe unter anderem von der Windlauflänge abhängt. Zum anderen sind selbst

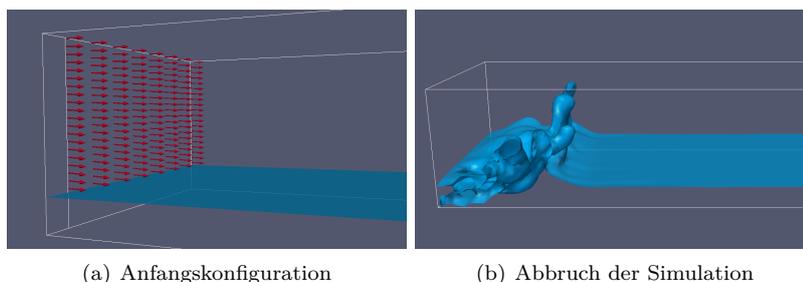


Abb. 4.1.1: Einer der Versuche zur Wellenerzeugung durch Wind. Auf der linken Seite ist die Windgeschwindigkeit durch Pfeile dargestellt; auf der rechten Seite ist die Wasseroberfläche kurz vor Abbruch der Simulation zu sehen (eigene Abbildung).

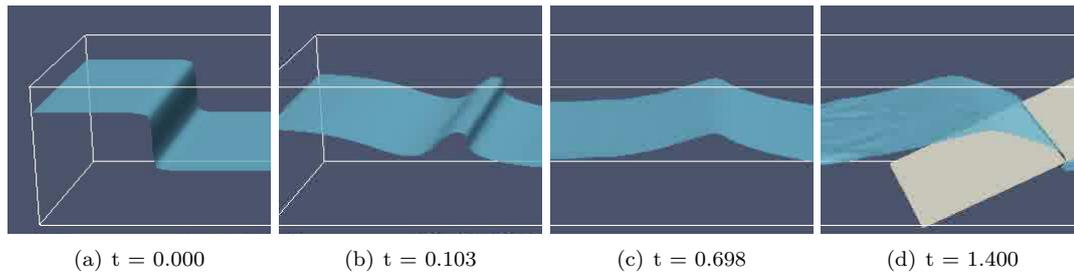


Abb. 4.1.2: Erzeugung einer einzelnen Welle durch eine fallende Wassersäule. Die Bilderfolge veranschaulicht den Fortschritt der Welle bis zum Aufprall auf die gegenüber liegende Schräge. Jedes Bild zeigt einen anderen Abschnitt des Beckens zum Zeitpunkt t (eigene Abbildung).

dann nicht unbedingt realistische Hochseewellen zu erwarten, da die Luft in der Simulation als inkompressibel angenommen wird.

Eine andere Möglichkeit eine Welle zu erzeugen bietet die bereits erwähnte fallende Wassersäule. Hierfür wird die Level-Set Funktion so initialisiert, dass die Wasseroberfläche eine Art Stufe bildet. Durch die Schwerkraft fällt diese Wassersäule während der Simulation und erzeugt dadurch eine einzelne Welle. Dieses Verfahren ist durch den geringen Aufwand insbesondere für eine einzelne große Welle geeignet. Abbildung 4.1.2 zeigt eine auf diese Weise erzeugte Welle, sowie die Anfangskonfiguration der Wassersäule. Dadurch können nun zwar einzelne Wellen generiert werden, jedoch keine Abfolge von mehreren Wellen.

Alle durchgeführten Versuche haben gezeigt, wie schwierig es ist die Gegebenheiten in der Natur nachzuempfinden. Wind und die daraus resultierende Wellenentstehung sind äußerst komplex und nur schwer nachzubilden. Es gibt allerdings noch eine weitere Möglichkeit Wellen zu generieren. In Laboratorien werden Wellen im Allgemeinen durch mechanisch angetriebene Wellenerzeuger in der Form von Klappen, Kolben oder Ähnlichem erzeugt. Dazu werden lange Wellenbecken angelegt, an deren Ende ein solcher Wellenerzeuger durch regelmäßige Bewegungen das Wasser in Bewegung versetzt und auf diese Weise einen gleichmäßigen Wellengang erzeugt. Die Größe und Periode der Wellen hängt somit von diesen einfach steuerbaren Komponenten ab. Die Idee ist nun, einen solchen Versuchsaufbau mit Hilfe des in Abschnitt 3.5 beschriebenen Strömungslösers zu simulieren.

4.2 Der Versuchsaufbau

An dieser Stelle werden alle für die Simulation benötigten Komponenten beschrieben. Zunächst einmal wird ein **Wellenbecken**, das heißt ein zum Teil mit Wasser gefülltes Becken, benötigt. Das entspricht einem Zweiphasensystem, bei dem eine Phase Wasser und die andere Phase Luft ist. Das Becken ist dabei das Simulationsgebiet Ω . Dieser Teil kann bereits mit dem in Abschnitt 3.5 vorgestellten Strömungslöser berechnet werden.

Außerdem wird einen **Wellenabsorber** benötigt. Dieser besteht in den hier durchgeführten Simulationen — wie bereits auf Abbildung 2.3.1 angedeutet — aus einer einfachen Anhöhe mit der Steigung 1:4. Diese wird aus naheliegenden Gründen in der Literatur oft als *Strand* bezeichnet. Durch Bodenhaftung werden die ankommenden Wellen abgebremst und die Veränderung der Wassertiefe führt unter Umständen auch zum Brechen der Wellen, wie in Abschnitt 2.1 beschrieben. In Folge dessen geht ein Großteil der Wellenenergie verloren, wodurch die Wellen größtenteils absorbiert anstatt reflektiert werden. Das vermindert eventuell auftretende Interferenzen am Ende des Beckens und ermöglicht einen gleichmäßigen Wellengang. Dieser Wellenabsorber ist also ein einfacher unbeweglicher Festkörper. Wie genau dieser in den Strömungslöser eingebracht werden kann, wird in Abschnitt 4.3 beschrieben. Später wird der Wellenabsorber durch die zu untersuchenden Wellenkraftwerke ersetzt, die sich ebenfalls nicht bewegen.

Zu guter Letzt fehlt noch das Kernstück eines Wellenbeckens: der **Wellenerzeuger**. Dazu wird ein beweglicher Festkörper benötigt, dessen Bewegungen gezielt gesteuert werden können. Das kann mit dem bisher vorgestellten Strömungslöser nicht simuliert werden. Daher muss das Verfahren um eine *Fluid-Struktur-Wechselwirkung* erweitert werden. Die genaue Umsetzung dazu findet sich in Abschnitt 4.4 und 4.5.

4.3 Starrkörper Darstellungen auf dem Gitter

In diesem Abschnitt wird beschrieben, wie sich nicht-bewegliche Objekte in das Simulationsgebiet integrieren lassen. Zunächst einmal wird dazu jeder Gitterzelle ein Wert zugeordnet, je nachdem, ob die Zelle ein Fluid (*fluid*) oder Feststoff (*solid*) enthält. Außerdem werden auf den Rändern der Gitterzelle verschiedene Werte gesetzt, abhängig davon, ob sie an eine Gitterzelle mit einem Fluid oder einem Feststoff grenzen. Auf diese Weise wird ein neuer Rand zwischen Fluid und Festkörper innerhalb des Gebietes definiert, auf dem wiederum Randbedingungen gesetzt werden müssen. In diesem Fall wird eine Hafttrandbedingung auf dem Wellenabsorber gesetzt, damit die Wellen möglichst gut aufgehalten werden.

Da diese Art der Strukturdarstellung auf dem verwendeten Diskretisierungsgitter basiert, ist die Darstellung des Objektes maximal so fein aufgelöst wie das Gitter selbst. In Abbildung 4.2.1 ist das Diskretisierungsgitter deutlich zu sehen. Daher sind komplexe Strukturen nur auf sehr feinen Gittern gut darstellbar. Einfache Formen hingegen stellen kein großes Problem dar. Ein typisches Beispiel für solche festen Starrkörper ist eine Strömung um ein Hindernis in dem Fluidgebiet, wie beispielsweise ein Leuchtturm im Meer. Auch der Wellenabsorber kann auf diese Weise dargestellt werden.



Abb. 4.2.1: Darstellung einer Sphäre als sogenanntes *Flagfeld*. Dabei ist deutlich die verwendete Gitterstruktur zu erkennen (eigene Abbildung).

4.4 Komplexe Starrkörper

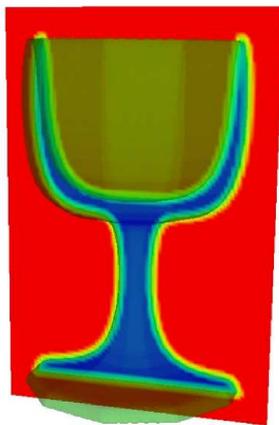


Abb. 4.4.1: Darstellung der transparenten Isofläche der Starrkörper Level-Set Funktion ϕ_s und eine Schnittebene, die die Abstandswerte mit unterschiedlichen Farben markiert (aus [Cro10]).

Es gibt viele Möglichkeiten komplexe Starrkörpergeometrien auf einem Gitter darzustellen. Eine davon wurde im vorherigen Abschnitt beschrieben. Eine solche an das Diskretisierungsgitter gebundene Struktur ist allerdings problematisch, wenn diese bewegt werden soll, da die grobe Darstellung keine feinen Bewegungen zulässt.

Ein Ansatz besteht nun darin einen solchen beweglichen Festkörper durch eine Level-Set Funktion darzustellen. Dies ermöglicht eine wesentlich genauere Bewegung der Struktur und hat zudem noch gute Eigenschaften hinsichtlich der nötigen Randbedingungen.

Da es nur um die Geometrie des zu bewegenden Objektes geht, wird die Level-Set Funktion ϕ_s des Starrkörpers lediglich in einem bestimmten Abstand um das Objekt herum berechnet; auf dem restlichen Gebiet wird ein fester Wert gesetzt. Eine Veranschaulichung der verschiedenen Abstandswerte — hier wird nicht zwischen zwei Fluidphasen, sondern zwischen Innen und Außen unterschieden — ist in Abbildung 4.4.1 zu sehen. Für genaue Details zur Berechnung von ϕ_s aus einer gegebenen Triangulierung sei auf [Cro10] verwiesen.

Die Übergabe einer Geometrie an den Strömungslöser erfolgt über Konfigurationsdateien, die alle wichtigen Parameter und Strukturen enthalten. Ein beweglicher Festkörper wird ähnlich wie die oben beschriebenen nicht-beweglichen Festkörper als Polygon beschrieben. Das heißt es werden Punkte und Kanten übergeben, die den Körper beschreiben. Im Unterschied zu einer unbeweglichen Geometrie müssen dabei noch die beiden Makros `lsgeom` und `DynamicLSGeometryUni` angegeben werden. Ein Stempel-Wellenerzeuger wird beispielsweise durch

```

poly {
    // Polygon
    lsgeom // Level-Set Geometrie
    DynamicLSGeometryUni // Uni-direktionale Kopplung
    points 8,
        <0.09,0.04,0.04>, // Punkt 0
        <0.10,0.04,0.04>, // Punkt 1
        <0.10,0.30,0.04>, // Punkt 2
        <0.09,0.30,0.04>, // Punkt 3
        <0.09,0.30,0.26>, // Punkt 4
        <0.10,0.30,0.26>, // Punkt 5
        <0.10,0.04,0.26>, // Punkt 6
        <0.09,0.04,0.26> // Punkt 7
    vertices 36,
        0,1,2,3,0,-1, // Kante 1
        1,2,5,6,1,-1, // Kante 2
        2,3,4,5,2,-1, // Kante 3
        4,5,6,7,4,-1, // Kante 4
        0,1,6,7,0,-1, // Kante 5
        0,3,4,7,0,-1 // Kante 6
}

```

beschrieben. Dabei sind die Punkte als Koordination in $\langle x, y, z \rangle$ -Richtung angegeben und die Kanten als geschlossene Kantenzüge zwischen den in der entsprechenden Reihenfolge angegebene Knoten, wobei -1 den jeweiligen Kantenzug abschließt. Die später untersuchten Wellenkraftwerke werden ebenfalls als Polygone beschrieben.

Beim Transport des Starrkörpers ist insbesondere darauf zu achten, dass die Deformationsfreiheit erhalten bleibt. Das heißt, dass sich der Körper nicht verformen darf. Daher eignen sich zum Transport der Starrkörper Level-Set Funktion Semi-Lagrange Diskretisierungen, die im Gegensatz zu Eulerschen Verfahren weniger numerische Diffusion aufweisen und somit die Starrheit des Körpers nur wenig beeinflussen. Außerdem geht die Starrkörpergeschwindigkeit \mathbf{u}_s durch einen Translations- und einen Rotationsanteil in die Impulsgleichungen ein. Wie die Kopplung zwischen Festkörper und Fluid erfolgt, wird in den folgenden Abschnitten beschrieben.

4.5 Fluid–Struktur–Interaktion

Wie das Wort *Interaktion* bereits andeutet, sind dabei nicht nur die Wirkung des Starrkörpers auf das Fluid wichtig, sondern auch umgekehrt die Wirkung des Fluids auf den Starrkörper. Dementsprechend wird zwischen einem *uni-* und *bidirektional gekoppelten* System unterschieden. Bei unidirektionaler Kopplung werden Translations- und Rotationsgeschwindigkeit vorgegeben, während bei bidirektionaler Kopplung Translation und Rotation aus der Strömungsdynamik des Fluids berechnet werden.

Für die späteren Simulationen werden lediglich Festkörper benötigt, bei denen die Geschwindigkeit gesteuert, das heißt fest vorgegeben, werden kann. Daher ist hier nur die unidirektionale Kopplung von Interesse. Croce [Cro10] beschreibt in seiner Arbeit beide Richtungen ausführlich.

4.5.1 Die Modellgleichungen

Für das erweiterte Modell mit beweglichen Starrkörpern sind einige neue Gleichungen von Interesse. Diese werden an dieser Stelle aufgeführt.

Zu Beginn muss das Gebiet Ω weiter unterteilt werden, um zwischen Fluid und Starrkörper zu unterscheiden. Dazu werden die Bezeichnungen

- \mathcal{F} für das Teilgebiet, das ausschließlich Fluide enthält,
- \mathcal{R} für das Teilgebiet, das vom Starrkörper eingenommen wird und
- $\partial\mathcal{R}$ für den gemeinsamen Rand beider Teilgebiete

eingeführt. Außerdem werden die vorher eingeführten Bezeichnungen für die Fluidteilgebiete benutzt. Somit kann nun ein Fluid–Starrkörper–System mit Erweiterung auf ein Zweiphasensystem dargestellt werden. Dazu werden folgende Gleichungen benötigt [Cro10]:

Fluid–Bewegungsgleichungen in 3D:

$$\rho_i \frac{d\mathbf{u}_i}{dt} = \rho_i \mathbf{g} + \nabla \cdot \mathbf{T}_i + \sigma \kappa \delta(\phi) \mathbf{n} \quad \text{in } \Omega_i \setminus \overline{\mathcal{R}(t)}, \quad (4.5.1a)$$

$$\nabla \cdot \mathbf{u}_i = 0 \quad \text{in } \Omega_i \setminus \overline{\mathcal{R}(t)}, \quad (4.5.1b)$$

$$\mathbf{u}_i = \mathbf{u}_\Gamma(t) \quad \text{auf } \Gamma, \quad (4.5.1c)$$

$$\mathbf{u}_i = \mathbf{v} + \mathbf{w} \times \mathbf{r} \quad \text{auf } \partial\mathcal{R}(t), \quad (4.5.1d)$$

$$\mathbf{u}_i|_{t=0} = \mathbf{u}_{i,0} \quad \text{in } \Omega_i \setminus \overline{\mathcal{R}(0)}. \quad (4.5.1e)$$

Kinematische Gleichungen in 3D:

$$\frac{d\mathbf{x}_s}{dt} = \mathbf{v} \quad \text{in } \mathcal{R}(t), \quad (4.5.2a)$$

$$\frac{d\boldsymbol{\theta}}{dt} = \mathbf{w} \quad \text{in } \mathcal{R}(t), \quad (4.5.2b)$$

$$\mathbf{x}_s|_{t=0} = \mathbf{x}_{s,0} \quad \text{in } \mathcal{R}(0), \quad (4.5.2c)$$

$$\boldsymbol{\theta}|_{t=0} = \boldsymbol{\theta}_0 \quad \text{in } \mathcal{R}(0). \quad (4.5.2d)$$

Dabei sind alle Bezeichnungen wie in Abschnitt 3.2, sowie \mathbf{x}_s der Schwerpunkt des Starrkörpers, \mathbf{v} die Translationsgeschwindigkeit, \mathbf{w} die Rotationsgeschwindigkeit und $\boldsymbol{\theta}$ die Winkelorientierung des Starrkörpers. Außerdem wird angenommen, dass der Festkörper sich nicht verformen lässt und die sich Bewegung aus Translation und Rotation zusammen setzt. Damit ergeben sich zusätzlich die beiden Gleichungen

$$\mathbf{D}[\mathbf{u}] = 0 \quad \text{in } \mathcal{R}(t), \quad (4.5.3a)$$

$$\mathbf{x}' = \mathbf{v} + \mathbf{w} \times \mathbf{r} \quad \text{in } \mathcal{R}(t), \quad (4.5.3b)$$

wobei $\mathbf{D}[\mathbf{u}] = [\nabla \mathbf{u} + \nabla \mathbf{u}^T]$ der Deformationstensor ist und daher Gleichung (4.5.3a) die Deformationsfreiheit garantiert. $\mathbf{x} \in \mathcal{R}$ ist ein beliebiger Punkt innerhalb des Starrkörpers und $\mathbf{r} := \mathbf{x} - \mathbf{x}_s$ ist der Richtungsvektor vom Schwerpunkt des Starrkörpers zu \mathbf{x} .

Ferner werden ausschließlich Starrkörper mit homogener Dichte $\rho_s = \textit{konstant}$ untersucht, wodurch sich die Masse durch $\rho_s \int_{\mathcal{R}} 1 \, d\mathbf{x}$ ergibt.

Um die Bewegung des Starrkörpers nun in das Zweiphasensystem einzubinden, werden die Impulsgleichungen und die Kontinuitätsgleichung entsprechend verändert.

4.5.2 Einbindung des Starrkörpers in das Zweiphasenmodell

Nun muss der Starrkörper durch die im vorherigen Abschnitt angegebenen Gleichungen in das Zweiphasensystem eingebunden werden. Dies geschieht über die Stoffeigenschaften — Dichte und Viskosität — sowie die Geschwindigkeit und die Level–Set Funktion des Festkörpers.

Da die Level–Set Funktion, die den Starrkörper beschreibt, wieder einen Sprung an der Grenze zwischen Körper und Fluid aufweist, muss diese für die Diskretisierung geglättet werden. In [Cro10] wird dazu eine angepasste Formulierung der *Smoothed Profile Method (SPM)* [NY05] verwendet. Hierfür wird die Gesamtgeschwindigkeit \mathbf{u} als glatte Konvexkombination der Starrkörpergeschwindigkeit \mathbf{u}_s und Fluidgeschwindigkeit \mathbf{u}_f dargestellt.

Sei nun ϕ_s die Level–Set Funktion der Starrkörpergeometrie und $H^\epsilon(\phi_s)$ die zugehörige geglättete Heaviside Sprungfunktion, wie in Gleichung (3.5.2) definiert. Dann lässt sich die Starrkörpergeschwindigkeit berechnen durch

$$H^\epsilon(\phi_s) \mathbf{u}_s(\mathbf{x}, t) = H^\epsilon(\phi_s) (\mathbf{v}(t) + \mathbf{w}(t) \times \mathbf{r}) \quad \text{in } \mathcal{R} + \epsilon, \quad (4.5.4)$$

wobei ϵ den Radius beschreibt, innerhalb dessen der Starrkörper durch die Level–Set Funktion ϕ_s beschrieben wird. Das totale Geschwindigkeitsfeld setzt sich schließlich aus Starrkörpergeschwindigkeit \mathbf{u}_s und Fluidgeschwindigkeit \mathbf{u}_f zusammen, sodass die glatte Konvexkombination

$$\begin{aligned} \mathbf{u} &:= (1 - H^\epsilon(\phi_s)) \mathbf{u}_f + H^\epsilon(\phi_s) \\ &= \mathbf{u}_f + H^\epsilon(\phi_s) (\mathbf{u}_s - \mathbf{u}_f) \end{aligned} \quad (4.5.5)$$

entsteht. Außerdem müssen noch zwei Bedingungen an das neue System gestellt werden. Eine davon ist die *Hafttrandbedingung* auf dem Rand des Starrkörpers, das heißt es soll $\mathbf{u}_s = \mathbf{u}_f$ auf $\partial\mathcal{R}$ gelten.

Schließlich wird noch — wie auch für das Zweiphasensystem — die Divergenzfreiheit des totalen Geschwindigkeitsfeldes \mathbf{u} gefordert, wobei diese in dem vorliegenden Fall die beiden Phasen sowie den Starrkörper betrifft. Unter Divergenzfreiheit ergibt sich aus Gleichung (4.5.5) die *Impermeabilitätsbedingung* für den Starrkörper. Diese besagt, dass der Rand des Starrkörpers undurchlässig ist, das heißt es muss $(\nabla H^\epsilon(\phi_s)) \cdot (\mathbf{u}_s - \mathbf{u}_f) = 0$ gelten. Um diese Folgerung einzusehen, wird der Divergenzoperator auf das totale Geschwindigkeitsfeld angewendet und es ergibt sich

$$\begin{aligned} 0 &\stackrel{!}{=} \nabla \cdot \mathbf{u} \\ &= \nabla \cdot [\mathbf{u}_f + H^\epsilon(\phi_s)(\mathbf{u}_s - \mathbf{u}_f)] \\ &= (\nabla H^\epsilon(\phi_s)) \cdot (\mathbf{u}_s - \mathbf{u}_f). \end{aligned}$$

Aus Gleichung (4.5.5) soll nun das totale Geschwindigkeitsfeld — welches auch die Starrkörpergeschwindigkeit \mathbf{u}_s beinhaltet — in dem gesamten Gebiet Ω berechnet werden. Dazu werden die Navier–Stokes–Gleichungen für das Zweiphasensystem (3.2.1) formal um den Kraftdichteterm \mathbf{f}_s ergänzt. Dieser enthält die entsprechenden Terme für die Interaktion zwischen Starrkörper und Fluid und wird nicht explizit angegeben, da er sich implizit ergibt, siehe dazu Abschnitt 4.5.3. Damit ergeben sich die modifizierten Gleichungen für das Zweiphasensystem mit Starrkörper [Cro10]

$$\rho_i \frac{D\mathbf{u}_i}{Dt} = -\nabla p_i + \nabla \cdot (\mu_i \mathbf{D}_i) + \rho_i \mathbf{g} + \sigma \kappa \delta \mathbf{n} + \mathbf{f}_s \quad \text{in } \Omega_i, \quad (4.5.6a)$$

$$\nabla \cdot \mathbf{u}_i = 0 \quad \text{in } \Omega_i, \quad (4.5.6b)$$

$$\mathbf{u}_i|_\Gamma = 0 \quad \text{in } [0, T], \quad (4.5.6c)$$

$$\mathbf{u}_i|_{t=0} = \mathbf{u}_{0_i} \quad \text{in } \Omega_i, \quad (4.5.6d)$$

mit $i \in 1, 2$ und den Bezeichnungen aus Abschnitt 3.2. Außerdem ergeben sich für den Starrkörper die Gleichungen [Cro10]

$$\rho_s \frac{\partial \mathbf{u}_s}{\partial t} = -\nabla p_s + \rho_s \mathbf{g} \quad \text{in } \mathcal{R}, \quad (4.5.7a)$$

$$\nabla \cdot \mathbf{u}_s = 0 \quad \text{in } \mathcal{R}, \quad (4.5.7b)$$

wobei die Materialparameter Dichte ρ_s und Druck p_s mit eingehen und sich aus der geforderten Divergenzfreiheit implizit die Impermeabilitätsbedingung ergibt.

Die Idee des erweiterten Algorithmus besteht nun darin, die Starrkörpergeschwindigkeit \mathbf{u}_s über das zusätzliche Druckfeld p_s in die Hilfgeschwindigkeit u^* einzubinden. Der in Abschnitt 3.7 beschriebene Algorithmus muss dazu durch eine erweiterte Chorin–Temam–Projektionsmethode ergänzt werden.

4.5.3 Der erweiterte Algorithmus

Mit allen bis hierhin beschriebenen Formeln und Verfahren kann nun der genaue Ablauf des neuen Strömungslösers vorgestellt werden.

Zunächst wird das Integral für den Schwerpunkt des Starrkörpers mittels Mittelpunktregel berechnet, das heißt

$$\begin{aligned} \mathbf{x}_s^0 &= \int_{\Omega} H^\epsilon(\phi_s^0(\mathbf{x})) \mathbf{x} \, d\mathbf{x} \\ &\approx h^3 \sum_{i,j,k} H^\epsilon(\phi_s^0(\mathbf{x}_{i,j,k})) \mathbf{x}_{i,j,k}, \end{aligned} \quad (4.5.8)$$

wobei $\mathbf{x}_{i,j,k}$ die Zellmittelpunkte sind.

Anschließend werden Translations– und Rotationsgeschwindigkeit des Starrkörpers initialisiert,

$$\mathbf{v}^0 = \mathbf{v}_{init} \quad \text{und} \quad \mathbf{w}^0 = \mathbf{w}_{init}.$$

Nun wird wie in Algorithmus 1 ein Hilfgeschwindigkeitsfeld u^* berechnet, welches nun aber auch abhängig von ϕ_s ist,

$$u^* = u^n + \delta t [-\nabla \cdot (u^n \otimes u^n) + \mathbf{g} + \frac{1}{\rho(\phi^n, \phi_s^n)} (\nabla \cdot (\mu^\epsilon(\phi^n, \phi_s^n) \mathbf{D}^n) - (1 - H^\epsilon(\phi_s^n)) \sigma \kappa(\phi^n) \delta(\phi^n) \nabla \phi^n)], \quad (4.5.9)$$

wobei nun Dichte und Viskosität beide sowohl von ϕ^n als auch von ϕ_s^n abhängen.

Danach wird zuerst die Zweiphasengrenze ϕ^n transportiert, reinitialisiert und eventuell mittels Picard–Fixpunktiteration korrigiert. Dann wird auch die Starrkörper Level–Set Funktion mit der vorgegebenen Translations– und Rotationsgeschwindigkeit transportiert.

Da sich durch den Transport der Schwerpunkt des Starrkörpers möglicherweise verschoben hat, muss dieser neu berechnet werden. Dies geschieht analog zu Gleichung (4.5.8).

Aus dem Gleichungssystem

$$\frac{u^{*,n+1} - u^*}{\delta t} + \frac{\nabla p^{*,n+1}}{\rho(\phi^{n+1}, \phi_s^{n+1})} = 0, \quad (4.5.10a)$$

$$\nabla \cdot u^{*,n+1} = 0. \quad (4.5.10b)$$

ergibt sich durch Anwendung des Divergenzoperators auf Gleichung (4.5.10a) und einsetzen von Gleichung (4.5.10b) die *erste Druck–Poisson–Gleichung*

$$\nabla \cdot \left(\frac{\nabla p^{*,n+1}}{\rho^\epsilon(\phi^{n+1}, \phi_s^{n+1})} \right) = \nabla \cdot \frac{u^*}{\delta t}. \quad (4.5.11)$$

Das Lösen der Poisson–Gleichung (4.5.11) durch einen geeigneten Löser für das lineare Gleichungssystem liefert den gesuchten Druck $p^{*,n+1}$. Damit kann nun $u^{*,n+1}$ durch die erste Geschwindigkeitskorrektur

$$u^{*,n+1} = u^* - \frac{\delta t}{\rho^\epsilon(\phi^{n+1}, \phi_s^{n+1})} \nabla p^{*,n+1} \quad (4.5.12)$$

berechnet werden. In den vorläufigen Druck– und Geschwindigkeitsfeldern wird nicht zwischen Fluid und Starrkörper unterschieden, das heißt der Festkörper wird zunächst ebenfalls als Fluid betrachtet. Diese Annahme muss im Anschluss noch durch ein zusätzliches Druckfeld im Inneren des Starrkörpers korrigiert werden.

Im Fall der bidirektionalen Kopplung müssten an dieser Stelle verschiedene Kräfte berechnet werden, um die Starrkörpergeschwindigkeit aus der neuen Translations– und Rotationsgeschwindigkeit \mathbf{v}^{n+1} und \mathbf{w}^{n+1} zu erhalten. Da das hier verwendete Modell jedoch lediglich die unidirektionale Kopplung benötigt, werden \mathbf{v}^{n+1} und \mathbf{w}^{n+1} vorgegeben und die Starrkörpergeschwindigkeit ergibt sich durch

$$H^\epsilon(\phi_s^{n+1}) \mathbf{u}_s^{n+1}(\mathbf{x}, t) = H^\epsilon(\phi_s^{n+1})(\mathbf{v}^{n+1}(t) + \mathbf{w}^{n+1}(t) \times \mathbf{r}^{n+1}). \quad (4.5.13)$$

Anschließend kann das Hilfsdruckfeld p_s^* durch die *zweite Druck–Poisson–Gleichung*

$$\nabla \cdot \left(\frac{\nabla p_s^{*,n+1}}{\rho^\epsilon(\phi^{n+1}, \phi_s^{n+1})} \right) = \nabla \cdot \frac{H^\epsilon(\phi_s^{n+1})(u_s^{n+1} - u^{*,n+1})}{\delta t} \quad (4.5.14)$$

berechnet werden. Diese beinhaltet nun auch den Druck innerhalb des Starrkörpers und korrigiert die vorherige Betrachtung des Festkörpers als Fluid. Für diesen Schritt wird die Divergenzfreiheit vorausgesetzt, wodurch sich implizit die Impermeabilitätsbedingung ergibt. Somit kann schließlich die Starrkörpergeschwindigkeit \mathbf{u}_s in das Geschwindigkeitsfeld \mathbf{u} durch die zweite Geschwindigkeitskorrektur

$$u^{n+1} = u^{*,n+1} + H^\epsilon(\phi_s^{n+1})(u_s^{n+1} - u^{*,n+1}) - \frac{\delta t}{\rho^\epsilon(\phi^{n+1}, \phi_s^{n+1})} \nabla p_s^{*,n+1} \quad (4.5.15)$$

eingebunden werden. Durch diesen letzten Schritt wird die globale Divergenzfreiheit sowie die Starrheit des Körpers gesichert.

Der Druck kann schließlich einfach mittels

$$p^{n+1} = p^{*,n+1} + p_s^{*,n+1} \quad (4.5.16)$$

berechnet werden.

Analog zu Abschnitt 3.5.4, müssen noch Randbedingungen für die beiden Poisson–Probleme gestellt werden. Dazu wird jeweils eine Projektion auf die äußere Einheitsnormale des Gebietsrandes durchgeführt, um die beiden Neumann–Randbedingungen

$$\partial_n p^{*,n+1}|_{\Gamma_f} = \frac{\rho^\epsilon(\phi^{n+1}, \phi_s^{n+1})}{\delta t} (\mathbf{u}_\Gamma^* - \mathbf{u}_\Gamma^{n+1}) \cdot \mathbf{n}, \quad (4.5.17a)$$

$$\partial_n p_s^{*,n+1}|_{\Gamma_f} = \frac{\rho^\epsilon(\phi^{n+1}, \phi_s^{n+1})}{\delta t} (\mathbf{u}_\Gamma^* - \mathbf{u}_\Gamma^{n+1}) \cdot \mathbf{n} \quad (4.5.17b)$$

zu erhalten.

4.6 Der Algorithmus erweitert auf die Interaktion mit Starrkörpergeometrien

Abschließend für dieses Kapitel wird der gesamte Algorithmus für das Zweiphasensystem mit der Erweiterung auf die Interaktion mit Starrkörpern vorgestellt. Dieser läuft ebenso wie Algorithmus 1 parallel.

Algorithmus 2 Paralleler Löser für inkompressible Zweiphasensysteme mit beweglichen Starrkörpern (nach [Cro10])

Require: Gittermaschenweiten $\delta x, \delta y, \delta z \in \mathbb{R}$, Endzeit T , Level-Set Funktion ϕ_0 , Gebiet Ω

- 1: Setze die lokalen Teilgebiete $\Omega^q, q \in \{1, \dots, P\}$.
 - 2: Setze $t := 0, n := 0$ und $h := \max(\delta x, \delta y, \delta z)$.
 - 3: Initialisiere $u^n := u_0, p^n := p_0$ und $\phi^n := \phi_0$ auf allen Teilgebieten $\Omega^q, q \in \{1, \dots, P\}$.
 - 4: Setze Grenzschichtweite $\epsilon(\mathbf{x})$.
 - 5: Initialisiere Schwerpunkt \mathbf{x}_s^0 des Starrkörpers.
 - 6: Setze Randwerte für u^n auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 7: **Zeitschleife:**
 - 8: **while** $t \leq T$ **do**
 - 9: **Berechne Zeitschrittweite.**
 - 10: **Impulsgleichung:**
 Kommuniziere Randwerte für u^n und ϕ^n .
 Berechne Hilfgeschwindigkeitsfeld u^* nach Gleichung (4.5.9) auf Ω^q .
 Setze Randwerte für u^* auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 11: **Transportgleichung für die freie Oberfläche ϕ nach Gleichung (3.5.4).**
 - 12: **Reinitialisierung gemäß Gleichung (3.4.7).**
 - 13: **Picard-Fixpunktiteration (optional)**
 - 14: **Transportgleichung für den Starrkörper ϕ_s :**
 Transportiere die Level-Set Funktion ϕ_s .
 Berechne den Schwerpunkt des Starrkörpers ϕ_s^{n+1} .
 - 15: **Setze rechte Seite der Druck-Poisson-Gleichung I:**
 Kommuniziere Randwerte für u^* .
 Setze rechte Seite $b := \nabla \cdot u^*$ auf allen Teilgebieten.
 - 16: **Druck-Poisson-Gleichung I:**
 Berechne $p^{*,n+1}$.
 Kommuniziere Randwerte für p_{it}^* pro Iterationsschritt it .
 - 17: **Geschwindigkeitskorrektur I:**
 Berechne $u^{*,n+1}$ nach Gleichung (4.5.12) auf allen Teilgebieten.
 Setze Randwerte für $u^{*,n+1}$ auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 18: **Starrkörpergeschwindigkeit:**
 Berechne Starrkörpergeschwindigkeit \mathbf{u}_s^{n+1} nach Gleichung (4.5.13).
 - 19: **Setze rechte Seite der Druck-Poisson-Gleichung II:**
 Kommuniziere Randwerte für \mathbf{u}_s^* .
 Setze rechte Seite von Gleichung (4.5.14) auf allen Teilgebieten.
 - 20: **Druck-Poisson-Gleichung II:**
 Berechne $p^{*,n+1}$.
 Kommuniziere Randwerte für p_{it} pro Iterationsschritt it .
 - 21: **Geschwindigkeitskorrektur II:**
 Berechne u^{n+1} nach Gleichung (4.5.15) auf allen Teilgebieten.
 Setze Randwerte für u^{n+1} auf den lokalen Rändern $\partial\Omega^q \cap \partial\Omega$.
 - 22: Setze $t := t + \delta t$ und $n := n + 1$.
 - 23: **end while**
-

5 Simulationsergebnisse

In diesem Kapitel werden die Numerischen Ergebnisse verschiedener Tests beschrieben. Zum einen werden verschiedene Versuche zur Wellenerzeugung durchgeführt, um beliebige Wellen erzeugen zu können. Zum anderen werden diese erzeugten Wellen anschließend genutzt, um verschiedene OWC-Kraftwerke zu untersuchen.

Die Ergebnisse zeigen, dass Wellen mit beliebiger Wellenhöhe und Periode einwandfrei generiert werden können. Außerdem können mit Hilfe dieser erzeugten Wellen verschiedene Geometrien von Wellenkraftwerken in drei Raumdimensionen getestet werden.

5.1 Realisierung von Numerischen Wellenbecken

An dieser Stelle wird die Realisierung eines Numerischen Wellenbeckens mit dem zuvor beschriebenen Strömungslöser vorgestellt. Der Aufbau und die Funktionsweise wurden bereits in Abschnitt 2.3 beschrieben.

5.1.1 Das Becken

Zunächst muss die Größe des Wellenbeckens festgelegt werden. Dabei ist insbesondere die Länge in x -Richtung, also die Richtung, in welche sich die Wellen ausbreiten sollen, entscheidend. Einerseits brauchen die Wellenerzeuger einen gewissen Wirkungsbereich, andererseits muss auch der Wellenabsorber am anderen Ende des Beckens beachtet werden. Hinzu kommt noch, dass die Wellen erst in einer gewissen Entfernung von dem jeweiligen Wellenerzeuger vollständig ausgebildet sind. Bei einem Stempel-Wellenerzeuger beispielsweise beträgt diese Entfernung circa die dreifache Wassertiefe [DD90]. Bei einer Wassertiefe von 20 cm sind die Wellen demnach erst ab einer Entfernung von etwas 60 cm interessant.

In den ausgeführten Simulationen wird eine Wassertiefe von 20 cm und eine Gesamtlänge von 250 cm gewählt. Der Wellenabsorber besteht aus einer einfachen Schräge mit der Steigung $1 : 4$ und ist am Ende des Beckens 25 cm hoch. Dadurch werden die Wellen bereits vor dem Ende des Beckens aufgehalten und kaum reflektiert, da sie durch die Bodenhaftung an der Schräge an Energie verlieren.

In y -Richtung wird eine Höhe von 35 cm gewählt. Dies stellt sicher, dass die Wellen nicht über das Gebiet hinaus kommen. In z -Richtung wird eine Tiefe von 30 cm gewählt. Das gesamte Wellenbecken ist in Abbildung 5.1.1 dargestellt. Entlang der roten Linie wird jeweils die Wellenhöhe gemessen.

Insgesamt wird also ein Wellenbecken der Größe $250\text{ cm} \times 35\text{ cm} \times 30\text{ cm}$ mit einer Auflösung von einer Gitterzelle pro Zentimeter betrachtet. An allen Gebietsrändern wird eine Rutschbedingung gesetzt, um ein geschlossenes System ohne Masseverluste zu erhalten. Die Abmessungen des Wellenbeckens orientieren sich in etwa an an der Größe des in [HMB01] beschriebenen Wellenbeckens.



Abb. 5.1.1: Aufbau des implementierten Numerischen Wellenbeckens. Als Beispiel ist hier ein Keil Wellenerzeuger dargestellt. Das Becken ist $250\text{ cm} \times 35\text{ cm} \times 30\text{ cm}$ groß und an allen Gebietsrändern wird eine Rutschrandbedingung festgelegt. Entlang der roten Linie wird die Wellenhöhe gemessen (eigene Abbildung).

Um die verschiedenen Wellenerzeuger anschließend zu vergleichen wird die jeweilige Wellenhöhe in der Mitte des Beckens gemessen. Dort sind die Wellen bereits vollständig ausgebildet und treffen noch nicht auf den Wellenabsorber. Da die Wasseroberfläche als Nullniveaumenge der Level-Set Funktion ϕ dargestellt wird, wird die Wellenhöhe anhand dieser Funktion bestimmt. Dazu wird für festes $x = 125$ und festes $z = 15$ festgestellt, zwischen welchen Punkten (x, y_1, z) , (x, y_2, z) , $0 \leq y_1 < y_2 \leq y_{max} = 35$ ein Vorzeichenwechsel stattfindet. Anschließend wird zu dem Wert von y_1 der Absolutwert der Abstandsfunktion $|\phi(x, y_1, z)|$ addiert, um die Wellenhöhe entlang der Linie $\{(x, y, z) | 0 \leq y \leq y_{max}\}$ zu erhalten.

5.1.2 Die Wellenerzeuger

Wie bereits in Abschnitt 2.4 erwähnt, sollen nun vier verschiedene Wellenerzeuger implementiert werden. Diese sind in Abbildung 5.1.2 nochmals dargestellt. Stempel und Klappe sind dabei exakt gleich gebaut und unterscheiden sich ausschließlich durch die Art der Bewegung. Sie haben eine Höhe und Tiefe von jeweils 26 cm und sind 1 cm , das heißt eine Gitterzelle, breit. Die Höhe stellt sicher, dass der Wellenerzeuger zu jedem Zeitpunkt über die Wasseroberfläche hinaus ragt und somit maximalen Einfluss auf die Wellen hat. Ideal wäre es zwar, wenn der Wellenerzeuger bis zum Rand ginge, aber aufgrund der Implementierung muss ein gewisser Abstand zum Rand eingehalten werden, um eine korrekte Level-Set Darstellung des Starrkörpers zu gewährleisten. Dies hat allerdings nur einen geringen Einfluss auf die Wellenhöhe.

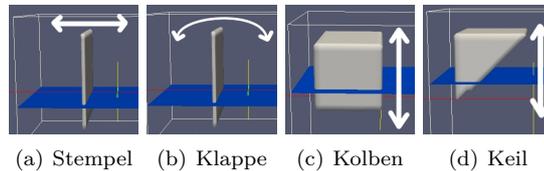


Abb. 5.1.2: Die vier verschiedenen Wellenerzeuger (eigene Abbildung).

Kolben und Keil haben ebenfalls einen ähnlichen Aufbau. Der Kolben ist im Querschnitt in der x - y -Ebene quadratisch, während der Keil im Querschnitt ein Dreieck ist. Dabei wird der Keil genau halb so groß gebaut, das heißt er entspricht einem diagonal halbierten Kolben. Die Größe dieser beiden Wellenerzeuger ist entscheidend für die Wellenhöhe und variiert deshalb in den späteren Experimenten.

Um einen regelmäßigen Seegang zu erhalten, werden verschiedene zeitabhängige Sinus-Funktionen als Geschwindigkeitsfunktionen für die Wellenerzeuger implementiert. Dadurch lässt sich neben der Geschwindigkeit auch die Wellenperiode beliebig steuern.

5.1.3 Die Kraftwerke

Eine Anwendung der im Rahmen dieser Arbeit erzeugten Wellen stellen Wellenkraftwerke dar. Wie bereits in Abschnitt 2.2 angesprochen, werden dazu *Oscillating-Water-Column-Kraftwerke* verwendet. Diese sind nicht beweglich und fest am Boden angebracht und sind aufgrund der geringen Wassertiefe als *Küsten-Wellenkraftwerke* zu verstehen, das heißt sie befinden sich direkt an der Küste — beispielsweise an einer Felswand oder werden als Wellenbrecher eingesetzt. Abbildung 5.1.3 gibt eine Übersicht über die verschiedenen Wellenkraftwerke, die in dieser Arbeit betrachtet werden.

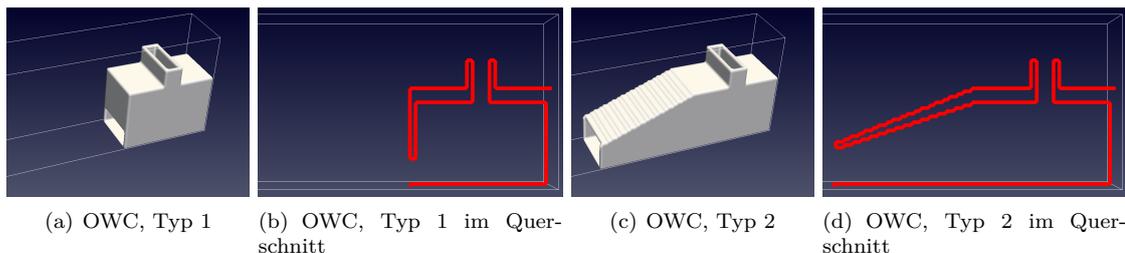


Abb. 5.1.3: Die verschiedenen OWC-Kraftwerke, die im Rahmen dieser Arbeit getestet werden (eigene Abbildung).

Tab. 5.1: Geschwindigkeiten der vier Wellenerzeuger in Abhängigkeit von der Zeit t . Die Translation geschieht dabei in x -Richtung und die Rotation um die z -Achse.

Versuch	Geschwindigkeit	Stempel	Klappe	Kolben	Keil
1	Translation (m/s)	$0.05 \sin(2\pi t)$	0	$0.04 \sin(4\pi t)$	$0.04 \sin(4\pi t)$
1	Rotation (Umdrehungen/s)	0	$2.5 \sin(2\pi t)$	0	0
2	Translation (m/s)	$0.07 \sin(2\pi t)$	0	$0.07 \sin(4\pi t)$	$0.07 \sin(4\pi t)$
2	Rotation (Umdrehungen/s)	0	$3.5 \sin(2\pi t)$	0	0
3	Translation (m/s)	$0.05 \sin(1.5\pi t)$	0	$0.04 \sin(4\pi t)$	$0.04 \sin(4\pi t)$
3	Rotation (Umdrehungen/s)	0	$2.5 \sin(1.5\pi t)$	0	0

5.2 Numerische Simulationen zur Wellenerzeugung

In diesem Abschnitt liegt der Fokus auf der Wellenerzeugung mit den in Abschnitt 2.4 beschriebenen Wellenerzeugern. Dazu wird eine Versuchsreihe angelegt, bei der die Bewegungsgeschwindigkeit der Wellenerzeuger, sowie deren Größe variiert. Daraus ergibt sich, dass sich die Wellenhöhe — wie zu erwarten — proportional zu dem Volumen des durch den Wellenerzeuger verdrängten Wassers verhält. Die Wellenperiode lässt sich durch die Geschwindigkeit des Wellenerzeugers beeinflussen und auch die Form der Welle hängt von der Form und der Bewegung des Starrkörpers ab.

Die Tabelle 5.1 gibt eine Übersicht über die jeweilige zeitabhängige Geschwindigkeit der Wellenerzeuger. Es wird jeweils sinus-Funktionen gewählt, um eine glatte und periodische Geschwindigkeitsfunktion zu erhalten. Durch die Glattheit wird die numerische Stabilität des verwendeten Algorithmus gesichert und die Periodizität sorgt für einen gleichmäßigen Wellengang. Dadurch wird der Vergleich der erzeugten Wellen erheblich vereinfacht.

Wie aus der Tabelle ersichtlich wird, werden in Versuch 1 und 3 für Kolben und Keil die gleichen Geschwindigkeiten verwendet, während in Versuch 2 höhere Geschwindigkeiten mit der gleichen Periode verwendet werden. Der Unterschied zwischen Versuch 1 und 3 liegt in der Größe der beiden Wellenerzeuger. In Versuch 3 ist das Volumen der Wellenerzeuger größer als in Versuch 1, wodurch mehr Wasser bei der Bewegung der Starrkörper verdrängt wird und so auch höhere Wellen zu erwarten sind. Für Stempel und Klappe ist es nicht sinnvoll das Volumen der Geometrien zu ändern. Stattdessen wird die Periode der Geschwindigkeitsfunktion vergrößert. Dadurch bleibt die Höchstgeschwindigkeit zwar gleich, doch sind die Intervalle mit niedrigeren Geschwindigkeiten dadurch größer. Somit sollten sich kleinere Wellen mit einer größeren Periode ergeben.

Durch die unterschiedlichen Funktionsweisen macht es Sinn, Stempel mit Klappe beziehungsweise Kolben mit Keil zu vergleichen.

Versuch 1: Moderate Geschwindigkeit

In diesem Versuch werden alle vier Wellenerzeuger eingesetzt, um Vergleichswerte für die späteren Versuche zu erhalten. Dabei erzeugen alle vier einen regelmäßigen Wellengang und halten auch Langzeitsimulationen Stand, ohne vor gegebener Endzeit abzubrechen.

Stempel und Klappe sind, wie in Abschnitt 5.1.2 beschrieben, gebaut, sodass sie zu jedem Zeitpunkt der Simulation aus dem Wasser herausragen und somit möglichst viel Volumen verschieben. Der Kolben hat eine Größe von $15 \times 15 \times 22$ Gitterzellen. Der Keil entspricht einem quer in der Mitte durchgeschnittenen Kolben und ist somit genau halb so groß.

Wie in Abbildung 5.2.1 zu sehen ist, erzeugen Stempel beziehungsweise Keil größere Wellen als Klappe beziehungsweise Kolben. Im ersten Fall von Stempel und Klappe ist dies nicht weiter verwunderlich, da der Stempel-Wellenerzeuger wesentlich mehr Wasser verdrängt als die Klappe. Da jedoch der Keil gerade mal halb so groß ist wie der der Kolben, scheint dieses Ergebnis zunächst überraschend. Die Erklärung dafür liegt in der Form. Da der Keil im Profil gesehen zur unteren Seite hin spitz zuläuft, wird das Wasser in die gewünschte Richtung, das heißt in Wellenaufrichtung, gedrückt. Beim Kolben hingegen ist auf beiden Seiten gleich viel Platz und das Wasser wird somit gleichmäßig zu allen Seiten weg geschoben. Auf diese Weise erzeugt der Keil höhere Wellen als der doppelt so große Kolben, da das Wasser hauptsächlich in Wellenaufrichtung gedrückt wird und daher eine größere in diese Richtung auf das Wasser wirkt.

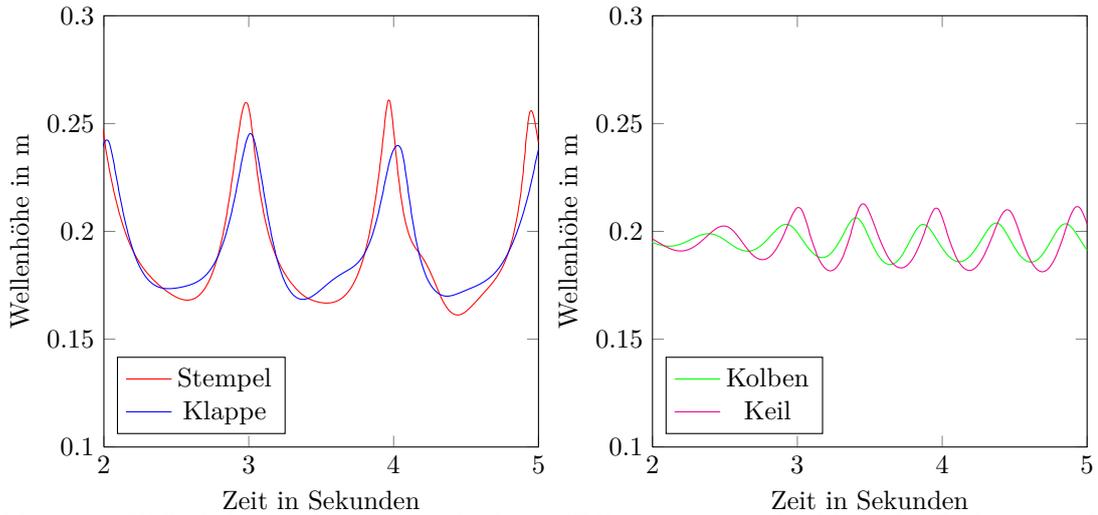


Abb. 5.2.1: Wellenhöhe für die vier verschiedenen Wellenerzeuger in der Mitte des Beckens mit den Geschwindigkeiten für Versuch 1 wie in Tabelle 5.1 dargestellt.

Versuch 2: Höhere Geschwindigkeit

In diesem Versuch werden die Geschwindigkeiten von allen vier Wellenerzeugern erhöht, ohne dabei die Periode zu verändern. Die Wellenerzeuger werden dabei nicht verändert.

Die mehr oder weniger geringen Auslenkungen der in Abbildung 5.2.2 dargestellten Funktionen lassen sich durch die Bewegung der Wellenerzeuger erklären. Da insbesondere bei Stempel und Klappe ein Sog entsteht, während diese sich zu ihrer Ursprungsposition zurück bewegen, entstehen dadurch ebenfalls kleinere Auslenkungen in der Wasseroberfläche. Bei Kolben und Keil hingegen entstehen die Unregelmäßigkeiten hauptsächlich durch das vollständige Auf- und Abtauchen der Wellenerzeuger. Aufgrund der höheren Geschwindigkeit sind daher in diesem Versuch größere Unebenheiten als in Versuch 1 zu erkennen. Dies ist allerdings kein Nachteil für die späteren Untersuchungen der Wellenkraftwerke, da reale Meereswellen ebenfalls nicht regelmäßig sind.

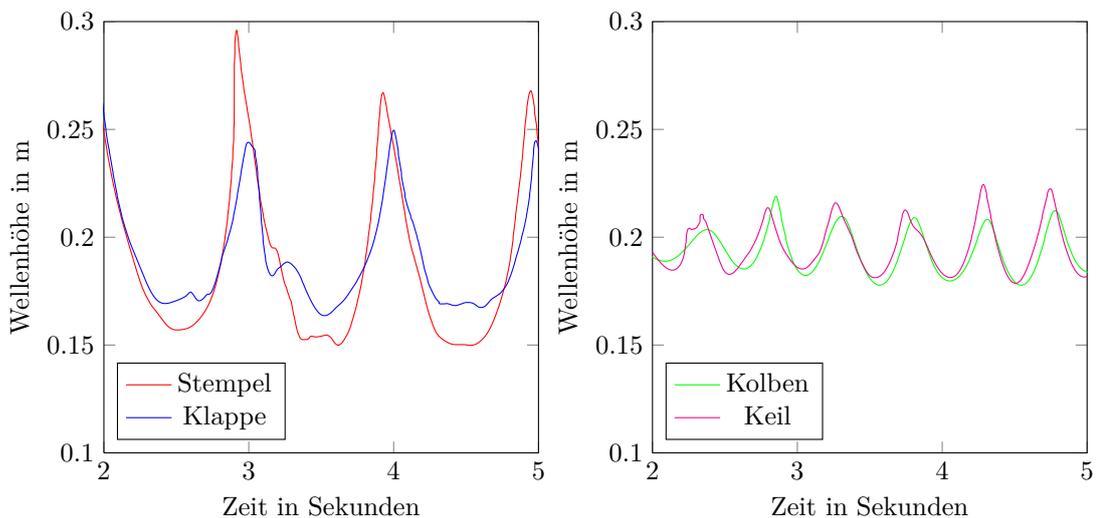


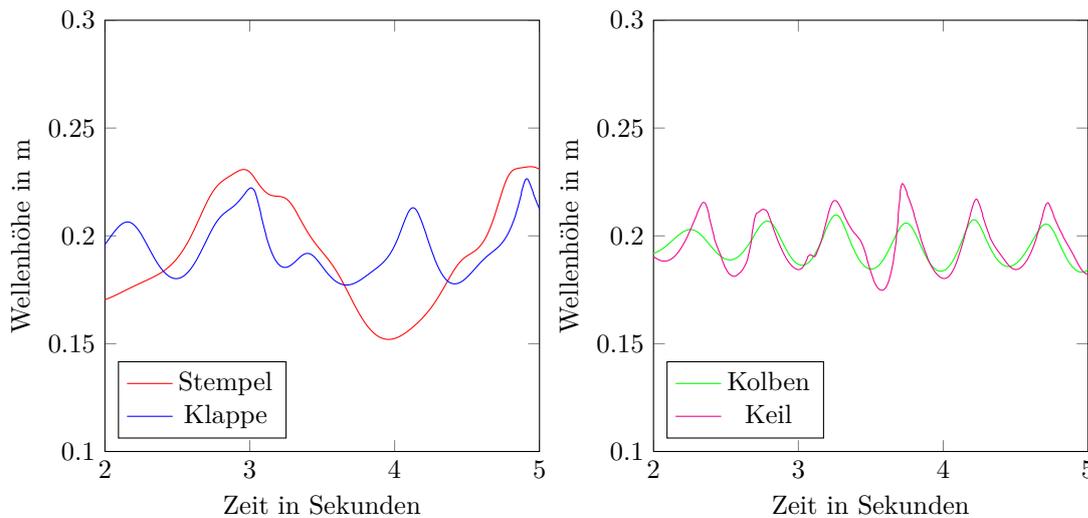
Abb. 5.2.2: Wellenhöhe für die vier verschiedenen Wellenerzeuger in der Mitte des Beckens mit den Geschwindigkeiten für Versuch 2 wie in Tabelle 5.1 dargestellt.

Tab. 5.2: Maximale Wellenhöhen in *cm* in der Mitte des Beckens für die vier verschiedenen Wellenerzeuger mit unterschiedlichen Parametern.

	Stempel	Klappe	Kolben	Keil
Versuch 1	26.168	24.535	20.630	21.274
Versuch 2	29.598	26.244	21.907	22.444
Versuch 3	23.593	22.648	20.970	22.425

Versuch 3: Größere Periode beziehungsweise größeres Volumen

In diesem Versuch werden die Geschwindigkeiten von Stempel und Klappe beziehungsweise die Volumen von Kolben und Keil so verändert, dass insgesamt ein größeres Volumen an Wasser verdrängt wird. Die Ergebnisse sind in Abbildung 5.2.3 dargestellt.

**Abb. 5.2.3:** Wellenhöhe für die vier verschiedenen Wellenerzeuger in der Mitte des Beckens mit den Geschwindigkeiten für Versuch 3 wie in Tabelle 5.1 dargestellt.

Hierbei wird zunächst für Stempel und Klappe ersichtlich, dass sich die Periode entsprechend der gewählten Geschwindigkeitsfunktion geändert hat. Bei Kolben und Keil hingegen ist diese erwartungsgemäß gleich geblieben. Außerdem sind wie erwartet bei Stempel und Klappe niedrigere Wellen und bei Kolben und Keil größere Wellen als in Versuch 1 zu beobachten.

Vergleich aller Versuche

Beim Vergleich aller drei Versuche ist insbesondere die erzielte Wellenhöhe entscheidend. Daher sind die maximalen Wellenhöhen der jeweiligen Versuche in Tabelle 5.2 aufgeführt. Aus der Tabelle wird deutlich, dass der Stempel-Wellenerzeuger die höchsten Wellen erzielt hat. Da dieser in der Lage ist, selbst bei einer moderaten Geschwindigkeit gleichmäßige und vergleichsweise hohe Wellen zu erzeugen, wird dieser Wellenerzeuger für die späteren Untersuchungen der Wellenkraftwerke verwendet.

Außerdem ist die Wellenperiode durch die Bewegung der Wellenerzeuger beeinflussbar, wie auch in den Abbildungen 5.2.1, 5.2.2 und 5.2.3 zu sehen ist. Dabei entspricht die Wellenperiode in etwa der Periode der Geschwindigkeitsfunktion der Wellenerzeuger.

Durch diese Testreihe konnte gezeigt werden, dass jede beliebige Art von Welle mit den verwendeten Wellenbecken erzeugt werden kann.

5.3 Numerische Simulationen zum Testen von Wellenkraftwerken

In diesem Abschnitt werden die verschiedenen OWC–Kraftwerke untersucht. Dazu wird ein Stempel–Wellenerzeuger genutzt und der erzeugte Luftstrom gemessen. Die verschiedenen Kraftwerke sind in Abbildung 5.1.3 dargestellt. Dabei wird eine relativ kurze Wellenperiode von einer Sekunde gewählt, um die Periode einer Schwerewelle (vgl. Abschnitt 2.1) zu erhalten und gleichzeitig mehrere Oszillationszyklen innerhalb der gewählten Laufzeit von zehn Sekunden zu beobachten.

Typ 1: Einfache Form

Das erste Kraftwerk in dieser Arbeit ist ein einfaches OWC–Kraftwerk. Der Aufbau entspricht einem Würfel, der an einer Seite eine Öffnung hat, um die ankommende Wellenenergie in das Innere zu lassen. Außerdem ist an der oberen Seite eine Öffnung, um die Luft ein– und ausströmen zu lassen. In dieser Öffnung wird der entstehende Luftstrom gemessen und mit den anderen Kraftwerken verglichen.

Typ 2: Wellenbrecher

Das zweite zu untersuchende Kraftwerk zeichnet sich durch eine Schräge aus, durch welche die ankommenden Wellen gebrochen werden. Das heißt, das erste Kraftwerk wurde so verändert, dass die den Wellen zugewendete Wand nicht mehr vertikal ist, sondern eine Schräge bildet. Solche Kraftwerke bieten sich insbesondere an Küsten an, um die dahinter liegenden Abschnitte vor der Kraft der Wellen zu schützen. Gleichzeitig wird dadurch die Resonanz des Wellenkraftwerkes beeinflusst, da die ankommenden Wellen weniger reflektiert werden als von einer simplen vertikalen Wand. Die Öffnung für die ankommenden Wellen sowie die Öffnung für den Luftstrom sind dabei ebenso groß wie beim ersten Kraftwerk. Allerdings ist nun das Innere der Kammer durch die zusätzliche Schräge etwas größer.

Vergleich der Kraftwerke

Zum Abschluss werden die verschiedenen Wellenkraftwerke verglichen. Dazu wird der induzierte Luftstrom in der Öffnung der OWC–Kammern untersucht. Abbildung 5.3.1 zeigt die durchschnittliche Geschwindigkeit der Luft in der Mitte der Öffnung. Um diese Werte zu messen, werden die Geschwindigkeitsvektoren auf einer Schnittebene parallel zur x – z –Ebene integriert und anschließend durch den Flächeninhalt der Schnittebene geteilt, um die Mittelwerte zu erhalten. Dabei sind bei beiden Kraftwerken die Öffnungen und die Schnittebene jeweils gleich groß, um die Kraftwerke vergleichen zu können.

Zunächst fällt die Oszillation der Geschwindigkeitswerte auf, welche das Ein– und Ausströmen der Luft verdeutlicht. Die Spitzengeschwindigkeit ist zwar beim ersten Kraftwerkstyp größer, doch weist der zweite Typ mehrere hohe Spitzenwerte auf. Da höhere Geschwindigkeiten der Luft eine größere Kraft auf die eingesetzte Turbine bewirken, sollte das zweite Kraftwerk etwas effizienter arbeiten als das erste. Dies hängt allerdings von der eingesetzten Turbine ab, die auch wiederum Einfluss auf den Luftstrom hat.

6 Zusammenfassung und Ausblick

Zusammenfassung

Im Rahmen dieser Arbeit wurden verschiedene Wellenerzeuger zur Simulation von Wellen in Numerischen Wellenbecken eingesetzt, um unterschiedliche Wellenkraftwerke zu testen. Dazu wurde erstmals ein dreidimensionaler Zweiphasen-Strömungslöser mit Fluid-Struktur-Interaktion und Level-Set Darstellung der freien Oberflächen eingesetzt.

Die erste Aufgabe bestand darin, realistische Wasserwellen zu erstellen. Die Erweiterung des Zweiphasenmodells auf die Fluid-Struktur-Interaktion ermöglicht es, Festkörper mit einer vorgeschriebenen Geschwindigkeit durch die zwei Phasen zu bewegen. Das durch diese Bewegung verdrängte Wasser generiert die gewünschten Wellen. Die Ergebnisse zeigen, dass jede beliebige Wellenhöhe und Periode realisierbar ist. Außerdem macht der Vergleich der eingesetzten Wellenerzeuger deutlich, dass der Stempel Wellenerzeuger bereits bei moderaten Geschwindigkeiten die größten Wellen generiert. Durch diesen vergleichsweise geringen Aufwand für die Wellengenerierung eignet sich daher der Stempel besonders gut für weitere Anwendungen.

Mit Hilfe der erzeugten Wellen ist es schließlich möglich, Wellenkraftwerke numerisch zu untersuchen. Angesichts des einfachen Grundmechanismus der sogenannten Oscillating-Water-Column Kraftwerke, bieten sich diese als Testobjekte an. Dazu wurden verschiedene Geometrien dieser Wellenenergiewandler konstruiert und analysiert. Da ein Zweiphasensystem betrachtet wird, konnte auch der entstehende Luftstrom untersucht werden. Dieser ist entscheidend für den Vergleich der unterschiedlichen Geometrien, da dieser für die Energieproduktion sorgt.

Ausblick

Die in dieser Arbeit durchgeführten Simulationen stellen eine vielversprechende Grundlage für weitere Studien zum Thema Wellenerzeugung und Wellenkraftwerke dar. Der Einsatz von anderen Wellenerzeugern, wie beispielsweise Zylindern oder Kugeln, könnte andere Wellenformen zu Stande bringen.

Außerdem gibt es weiterhin zahlreiche verschiedene Kraftwerke, die mit Hilfe der numerisch erzeugten Wellen untersucht werden können. Beispielsweise könnten auch schwimmende Wellenenergiewandler durch bidirektional gekoppelte Strukturen untersucht werden. Auch Resonanzeffekte zwischen auftreffenden Wellen und dem Kraftwerk könnten genauer untersucht werden. Ebenso wären vor dem Kraftwerk angebrachte Leitwände eine sinnvolle Erweiterung, um den dreidimensionalen Strömungslöser vollständig zu nutzen, da solche Kraftwerke in zwei Dimensionen nicht realisiert werden können. Ein anderer wichtiger Aspekt, gerade bei küstennahen Bauwerken, ist der Sedimenttransport. Von der Strömung transportierte Sedimentpartikel beeinflussen das Kraftwerk selber und insbesondere auch die Küstenform. Durch den Bau von Wellenkraftwerken könnte auf diese Weise das Sedimentbett und langfristig die gesamte Küstenform beeinflusst werden. Erfolgreiche Umsetzungen von strömungsbedingtem Sedimenttransport mit Hilfe des Strömungslösers *NaSt3DGP* gibt es bereits [Bur10].

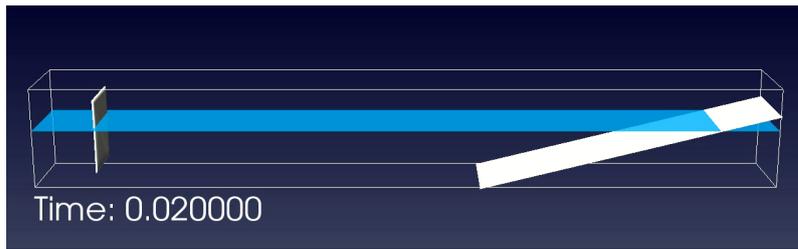
Insbesondere können durch die numerischen Simulationen neue Bauvorhaben vorab in numerischen Simulationen getestet und unter Umständen optimiert werden. Auf diese Weise können Kosten, Zeit und sehr viel Arbeitsaufwand gespart werden.

Insgesamt ergeben sich eine Reihe weiterführender faszinierender Fragen und Probleme, die in Zukunft beleuchtet werden können.

Anhang

An dieser Stelle werden ausgewählte Simulationsergebnisse dargestellt. Zur Erstellung der Bilder wurde die Visualisierungssoftware *ParaView* [Par] verwendet. Zu sehen sind jeweils Ausschnitte der Wellensimulation mit den vier verschiedenen Wellenerzeugern zu bestimmten Zeitpunkten. Diese entsprechen der 2. Versuchsreihe zur Wellenerzeugung aus Abschnitt 5.2.

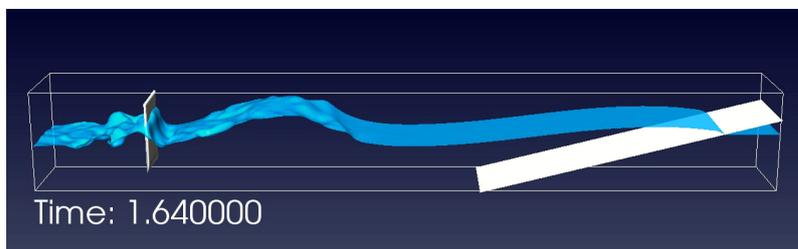
Außerdem sind noch die Simulationen der verschiedenen OWC-Kraftwerke abgebildet, ebenfalls jeweils zu bestimmten Zeitpunkten. Dabei sind auf der Schnittebene die Geschwindigkeitsvektoren farblich kodiert dargestellt. Die Kraftwerke sind im Querschnitt dargestellt, um die Kammer im Inneren visualisieren zu können.



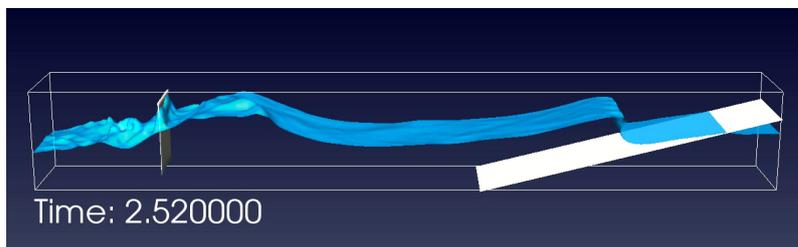
(a) Zeitpunkt $t = 0.00$



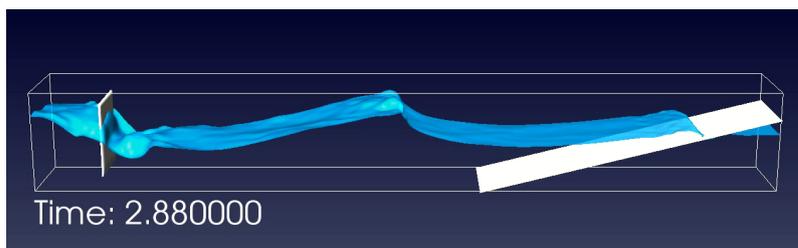
(b) Zeitpunkt $t = 0.84$



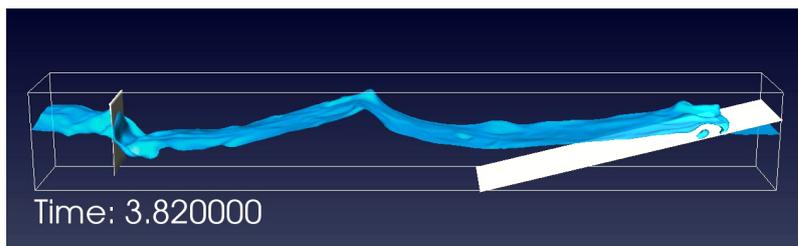
(c) Zeitpunkt $t = 1.64$



(d) Zeitpunkt $t = 2.52$

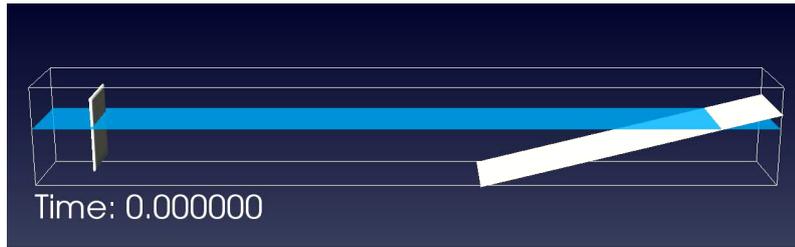


(e) Zeitpunkt $t = 2.88$



(f) Zeitpunkt $t = 3.82$

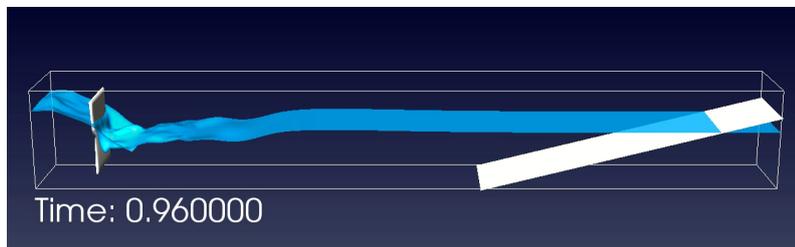
Abb. A1: Simulationsergebnisse mit einem Stempel Wellenerzeuger



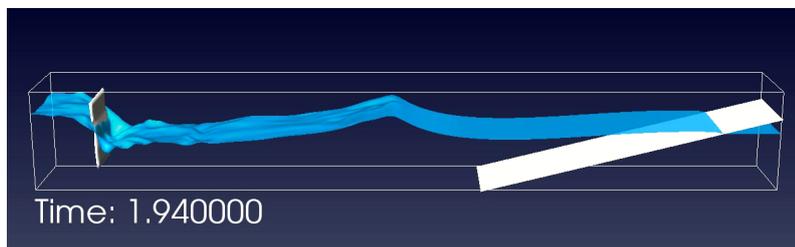
(a) Zeitpunkt $t = 0.00$



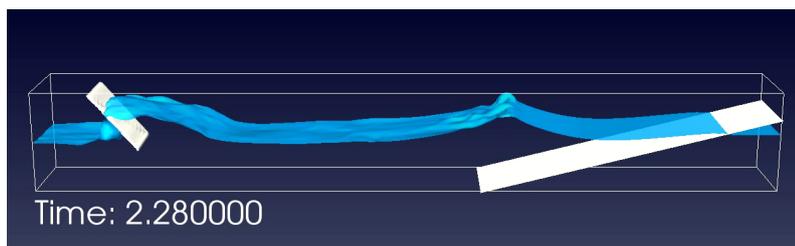
(b) Zeitpunkt $t = 0.76$



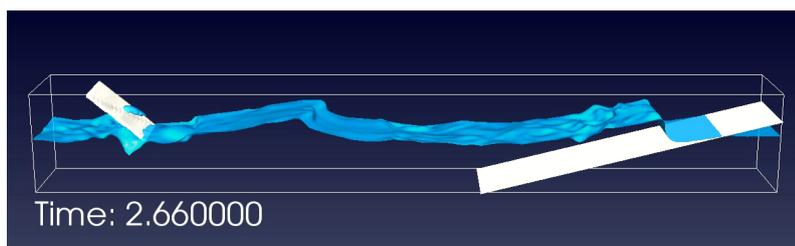
(c) Zeitpunkt $t = 0.96$



(d) Zeitpunkt $t = 1.94$

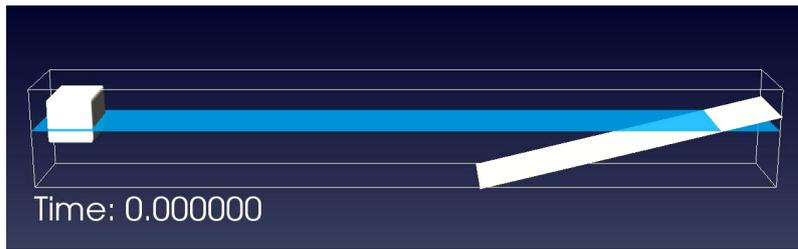


(e) Zeitpunkt $t = 2.28$

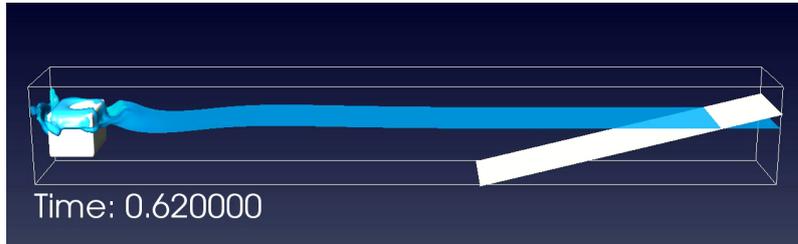


(f) Zeitpunkt $t = 2.66$

Abb. A2: Simulationsergebnisse mit einem Klappen Wellenerzeuger



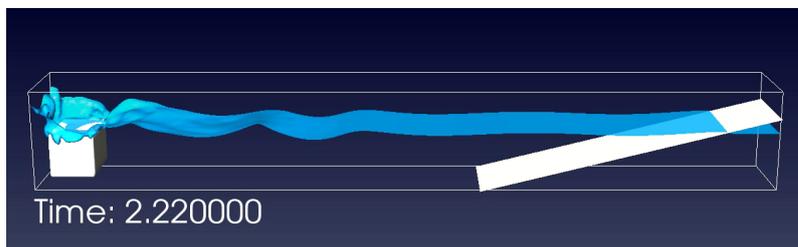
(a) Zeitpunkt $t = 0.00$



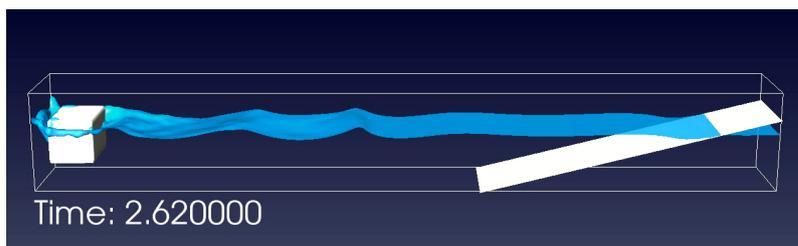
(b) Zeitpunkt $t = 0.62$



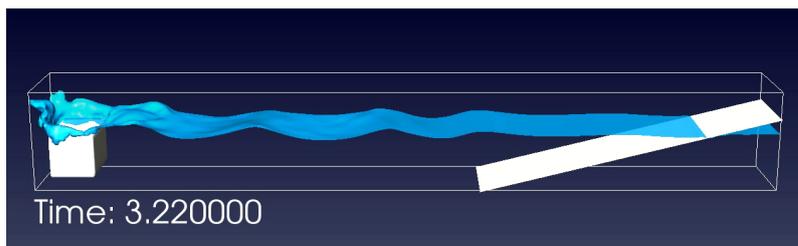
(c) Zeitpunkt $t = 1.04$



(d) Zeitpunkt $t = 2.22$

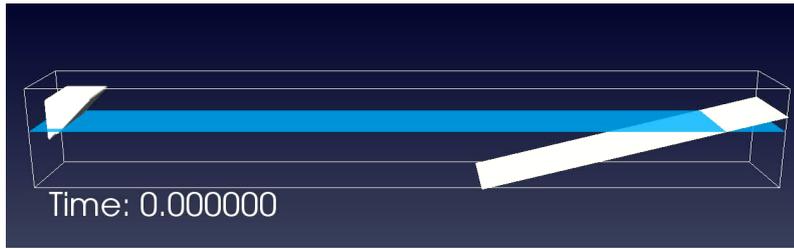


(e) Zeitpunkt $t = 2.62$

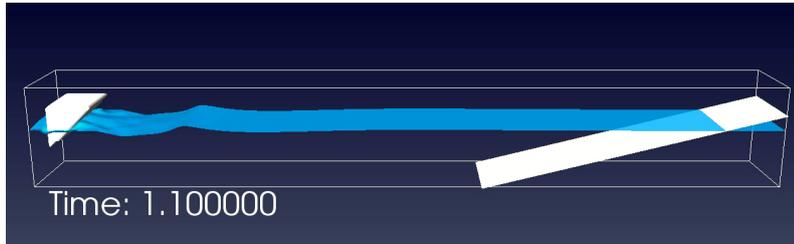


(f) Zeitpunkt $t = 3.22$

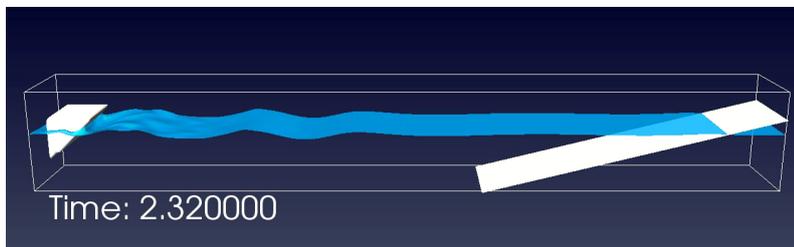
Abb. A3: Simulationsergebnisse mit einem Kolben Wellenerzeuger



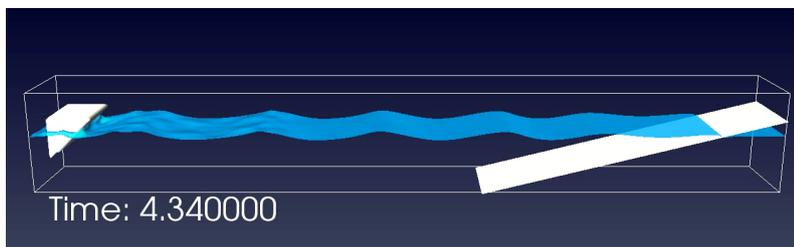
(a) Zeitpunkt $t = 0.00$



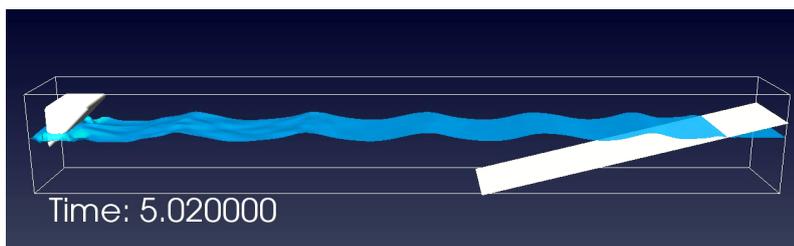
(b) Zeitpunkt $t = 1.10$



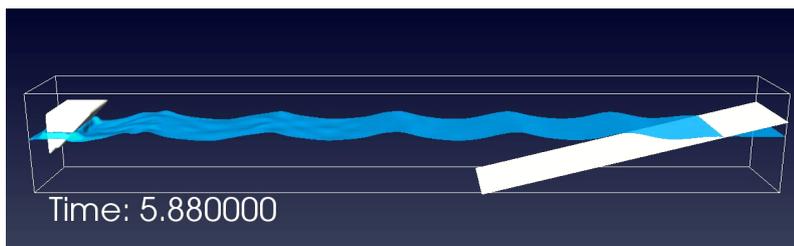
(c) Zeitpunkt $t = 2.32$



(d) Zeitpunkt $t = 4.34$

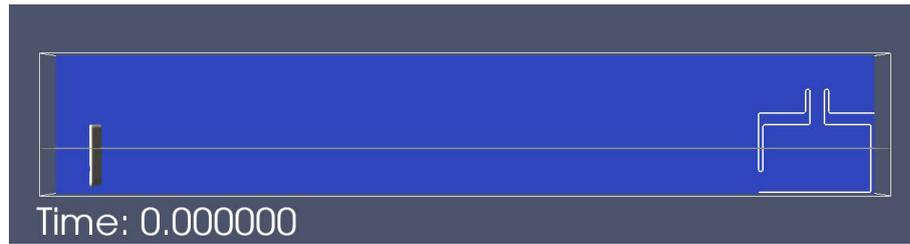


(e) Zeitpunkt $t = 5.02$

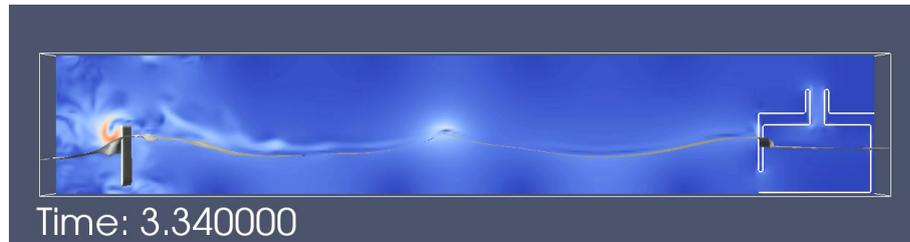


(f) Zeitpunkt $t = 5.88$

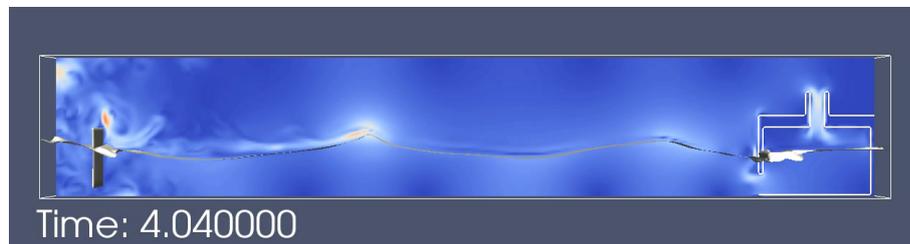
Abb. A4: Simulationsergebnisse mit einem Kolben Wellenerzeuger



(a) Zeitpunkt $t = 0.00$



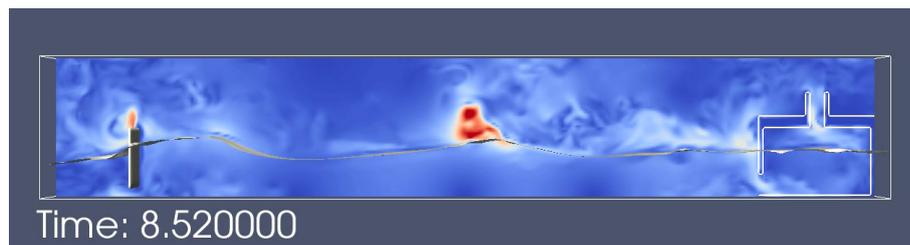
(b) Zeitpunkt $t = 3.34$



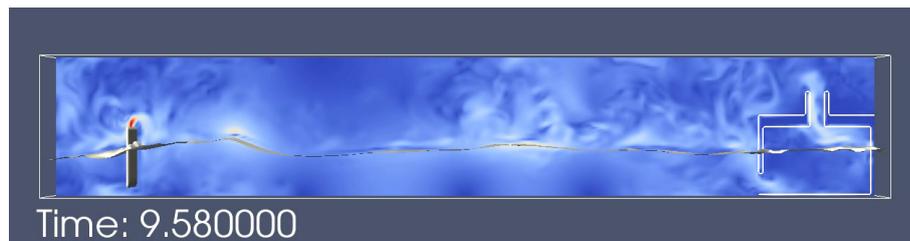
(c) Zeitpunkt $t = 4.04$



(d) Zeitpunkt $t = 7.03$

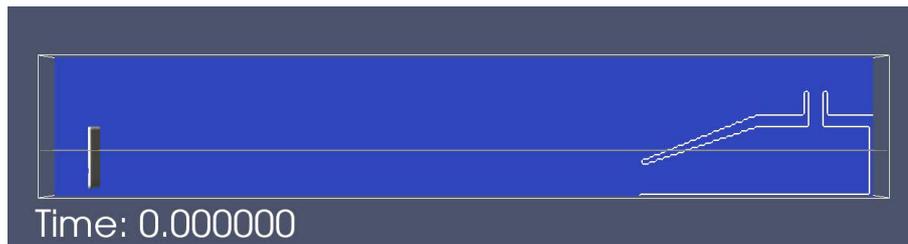


(e) Zeitpunkt $t = 8.52$

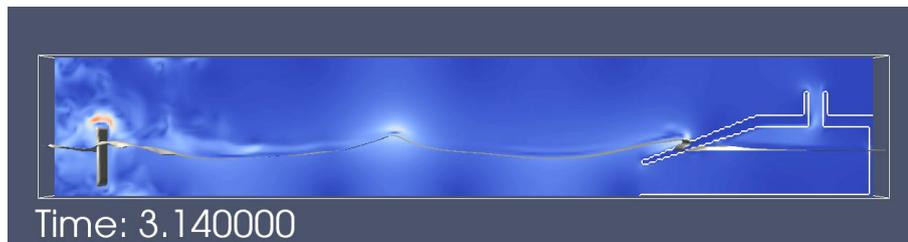


(f) Zeitpunkt $t = 9.58$

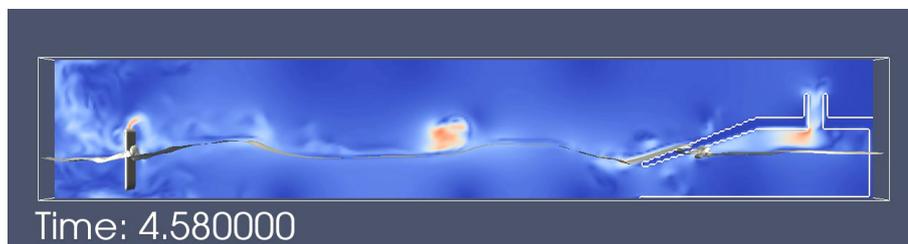
Abb. A5: Simulationsergebnisse des OWC Kraftwerkes Typ 1



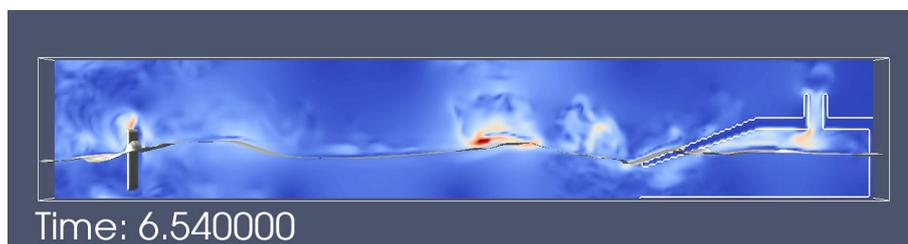
(a) Zeitpunkt $t = 0.00$



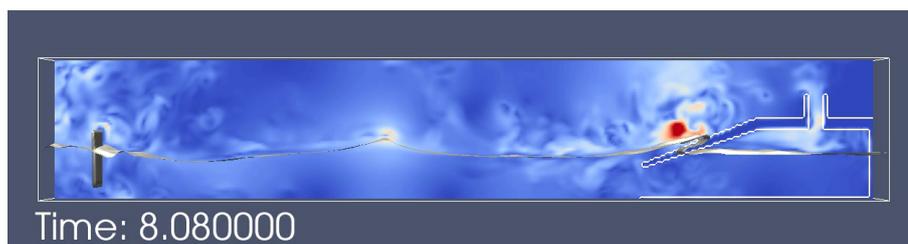
(b) Zeitpunkt $t = 3.14$



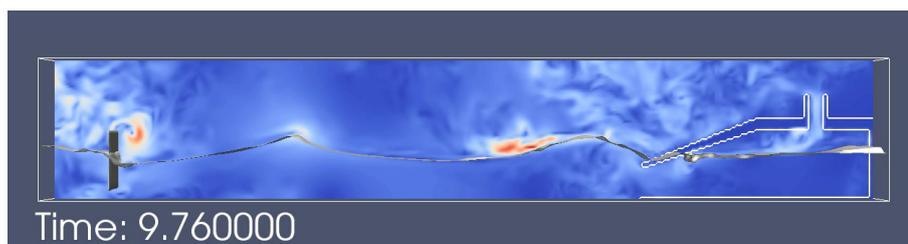
(c) Zeitpunkt $t = 4.58$



(d) Zeitpunkt $t = 6.54$



(e) Zeitpunkt $t = 8.08$



(f) Zeitpunkt $t = 9.76$

Abb. A6: Simulationsergebnisse des OWC Kraftwerkes Typ 2

Abbildungsverzeichnis

1.0.1 Skizze eines <i>Oscillating–Water–Column</i> Kraftwerkes. Die ankommenden Wellen bewirken eine vertikale Bewegung der Wassersäule im Inneren des Bauwerks. Dadurch wird die Luft im Inneren ebenfalls bewegt und treibt eine Turbine an (eigene Abbildung).	2
2.1.1 Verschiedene Wellenarten eingeteilt nach Frequenz beziehungsweise Periode, den auslösenden und den bestimmenden Kräften. Außerdem ist die geschätzte Energie der verschiedenen Wellen dargestellt (aus [Gra95]).	5
2.1.2 Einfluss der Wassertiefe auf die Wellen. Die Einteilung in Flachwasser, Tiefwasser und Übergangsbereich erfolgt über die relative Wassertiefe d/L . Die Ellipsen stellen die Bewegung der Wasserteilchen dar (aus [Gra95]).	6
2.2.1 Verschiedene Arten von Wellenkraftwerken. Die Einteilung erfolgt über die Bauweise (fest oder schwimmend, getaucht oder nicht getaucht) und die Funktionsweise (nach [Gra95]).	7
2.2.2 Pelamis Wellenkraftwerk. Die "Seeschlange" wird durch Hydraulikelemente zwischen den einzelnen Segmenten angetrieben (eigene Abbildung).	8
2.2.3 Funktionsweise eines <i>Oscillating–Water–Column</i> Kraftwerkes. Die bewegte Luftsäule im Inneren des Bauwerks treibt eine Turbine an, die an einen Generator angeschlossen ist. Auf diese Art wird Strom erzeugt (eigene Abbildung).	9
2.3.1 Typischer Aufbau eines Numerischen Wellenbeckens. An einen Ende befindet sich ein Wellenerzeuger und am anderen Ende ein Wellenabsorber, der ungewollte Wellenreflexionen reduziert (eigene Abbildung).	10
2.4.1 Schema eines Stempel–Wellenerzeugers im Flachwasser in der $x - z$ -Ebene. Dabei ist h die Wassertiefe, H die Wellenhöhe und S der Hubweg des Stempels (aus [DD90]).	11
2.4.2 Verschiedene Arten von Wellenerzeugern. Diese vier werden in dieser Arbeit zur Wellengenerierung eingesetzt (eigene Abbildung).	11
3.5.1 Eine Gitterzelle im versetzten Gitter. Die skalaren Größen liegen in den Mittelpunkten der Seitenflächen und die Vektoren im Mittelpunkt der Zelle (eigene Abbildung).	16
4.1.1 Einer der Versuche zur Wellenerzeugung durch Wind. Auf der linken Seite ist die Windgeschwindigkeit durch Pfeile dargestellt; auf der rechten Seite ist die Wasseroberfläche kurz vor Abbruch der Simulation zu sehen (eigene Abbildung).	20
4.1.2 Erzeugung einer einzelnen Welle durch eine fallende Wassersäule. Die Bilderfolge veranschaulicht den Fortschritt der Welle bis zum Aufprall auf die gegenüber liegende Schräge. Jedes Bild zeigt einen anderen Abschnitt des Beckens zum Zeitpunkt t (eigene Abbildung).	21
4.2.1 Darstellung einer Sphäre als sogenanntes <i>Flagfeld</i> . Dabei ist deutlich die verwendete Gitterstruktur zu erkennen (eigene Abbildung).	22
4.4.1 Darstellung der transparenten Isofläche der Starrkörper Level–Set Funktion ϕ_s und eine Schnittebene, die die Abstandswerte mit unterschiedlichen Farben markiert (aus [Cro10]).	22
5.1.1 Aufbau des implementierten Numerischen Wellenbeckens. Als Beispiel ist hier ein Keil Wellenerzeuger dargestellt. Das Becken ist $250\text{ cm} \times 35\text{ cm} \times 30\text{ cm}$ groß und an allen Gebietsrändern wird eine Rutschrandbedingung festgelegt. Entlang der roten Linie wird die Wellenhöhe gemessen (eigene Abbildung).	28
5.1.2 Die vier verschiedenen Wellenerzeuger (eigene Abbildung).	29
5.1.3 Die verschiedenen OWC–Kraftwerke, die im Rahmen dieser Arbeit getestet werden (eigene Abbildung).	29
5.2.1 Wellenhöhe für die vier verschiedenen Wellenerzeuger in der Mitte des Beckens mit den Geschwindigkeiten für Versuch 1 wie in Tabelle 5.1 dargestellt.	31

5.2.2	Wellenhöhe für die vier verschiedenen Wellenerzeuger in der Mitte des Beckens mit den Geschwindigkeiten für Versuch 2 wie in Tabelle 5.1 dargestellt.	31
5.2.3	Wellenhöhe für die vier verschiedenen Wellenerzeuger in der Mitte des Beckens mit den Geschwindigkeiten für Versuch 3 wie in Tabelle 5.1 dargestellt.	32
5.3.1	Durchschnittliche Geschwindigkeit des Luftstroms in der Öffnung für die Turbine. Die obere Grafik zeigt die Geschwindigkeiten für den ersten Kraftwerkstyp und die untere für den zweiten.	34
A1	Simulationsergebnisse mit einem Stempel Wellenerzeuger	37
A2	Simulationsergebnisse mit einem Klappen Wellenerzeuger	38
A3	Simulationsergebnisse mit einem Kolben Wellenerzeuger	39
A4	Simulationsergebnisse mit einem Kolben Wellenerzeuger	40
A5	Simulationsergebnisse des OWC Kraftwerkes Typ 1	41
A6	Simulationsergebnisse des OWC Kraftwerkes Typ 2	42

Literaturverzeichnis

- [BKZ92] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comput. Phys. **100** (1992), 335–354.
- [Bur10] M. Burkow, *Numerische Simulation stroemungsbedingten Sedimenttransports und der entstehenden Gerinnebettformen*, Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2010.
- [CGS09] R. Croce, M. Griebel, and M. A. Schweitzer, *Numerical Simulation of Bubble and Droplet-Deformation by a Level Set Approach with Surface Tension in Three Dimensions*, International Journal for Numerical Methods in Fluids **62** (2009), no. 9, 963–993, Also available as SFB 611 Preprint no 431.
- [CHMO96] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, *A level set formulation of Eulerian interface capturing methods for incompressible fluid flows*, J. Comput. Phys. **124** (1996), 449–464.
- [Cho68] A. Chorin, *Numerical Solution of the Navier-Stokes Equations*, Mathematics of Computation **22** (1968), 745–762.
- [CKZL99] G. Chen, C. Kharif, S. Zaleski, and J. Li, *Two-dimensional Navier-Stokes simulation of breaking waves*, Physics of Fluids **11** (1999), no. 1, 121–133.
- [Cro10] R. Croce, *Numerische Simulation der Interaktion von inkompressiblen Zweiphasenströmungen mit Starrkörpern in drei Raumdimensionen*, Dissertation, Rheinische Friedrich-Wilhelms-Universität Bonn, 2010.
- [CY09] J. Choi and S.B. Yoon, *Numerical simulations using momentum source wave-maker applied to RANS equation model*, Coastal Engineering **56** (2009), no. 10, 1043–1060.
- [DD90] R.G. Dean and R.A. Dalrymple, *Water wave mechanics for engineers and scientists*, Advanced Series on Ocean Engineering, vol. 2, ch. 6, pp. 170–185, World Scientific Publishing, 1990.
- [Fal02] J. Falnes, *Optimum control of oscillation of wave-energy converters.*, International Journal of Offshore and Polar Engineering **12** (2002), no. 2, 147–154.
- [GDN98] M. Griebel, T. Dornseifer, and T. Neunhoffer, *Numerical Simulation in Fluid Dynamics, a Practical Introduction*, SIAM, Philadelphia, 1998.
- [Gra95] K.-U. Graw, *Wellenenergie - eine hydromechanische Analyse*, Habilitation, Bergische Universität - Gesamthochschule Wuppertal, 1995.
- [Gra06] *Nutzung der Wellenenergie - Stand und Perspektiven*, 2006, 1. Deutsche Meeresenergie-Forum.
- [HG09] R.T. Hudspeth and R.B. Guenther, *Handbook of Coastal and Ocean Engineering*, ch. 2, pp. 25–56, World Scientific Publishing, 2009.
- [HMB01] P.J. Hudson, M.E. McCormick, and S.T. Browne, *A Low-Cost Wave-Sediment-Towing Tank*, 2001.
- [HN81] C.W. Hirt and B.D. Nichols, *Volume of fluid (vof) method for the dynamics of free boundaries*, Journal of computational physics **39** (1981), no. 1, 201–225.
- [HW+65] F.H. Harlow, J.E. Welch, et al., *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Physics of fluids **8** (1965), no. 12, 2182.
- [KCT99] C.H. Kim, A.H. Clement, and K. Tanizawa, *Recent Research and Development of Numerical Wave Tanks - A Review*, International Journal of Offshore and Polar Engineering **9** (1999), 241–256.

- [KfV⁺05] K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, and B. Buchner, *A Volume-of-Fluid based simulation method for wave impact problems*, Journal of Computational Physics **206** (2005), no. 1, 363–393.
- [Küf11] Georg Küffner, *Strom aus der Kraft der Wellen*, Frankfurter Allgemeine Zeitung, 28. Juli 2011.
- [LL99] P. Lin and P.L.-F. Liu, *Internal Wave-Maker for Navier-Stokes Equation Models*, Journal of Waterway, Port, Coastal, and Ocean Engineering **125** (1999), no. 4, 207–215.
- [ME88] P. McIver and DV Evans, *An approximate theory for the performance of a number of wave-energy devices set into a reflecting wall**, Applied ocean research **10** (1988), no. 2, 58–65.
- [Mik07] T. Mikkola, *Simulation of plunger type wave makers*, Journal of Structural Mechanics **40** (2007), no. 4, 19–39.
- [MPI] *Message Passing Interface (MPI)*, Website, <http://www.mcs.anl.gov/research/projects/mpi/>, letzter Zugriff 17.08.2011.
- [NY05] Yasuya Nakayama and Ryoichi Yamamoto, *Simulation method to resolve hydrodynamic interactions in colloidal dispersions*, Phys. Rev. E **71** (2005), no. 3, 036707.
- [OS88] Stanley Osher and James A Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics **79** (1988), no. 1, 12 – 49.
- [Par] *ParaView*, Website, <http://www.paraview.org>, letzter Zugriff 17.08.2011.
- [Pel] *Informationen über ein Projekt der Firma Pelamis Wave Power Ltd in Zusammenarbeit mit E.ON UK*, Website, <http://www.pelamiswave.com/our-projects/e-on-at-emec>, letzter Zugriff 17.08.2011.
- [PYM⁺01] J.C. Park, Y.Uno, H. Matsuo, T. Sato, and H. Miyata, *Reproduction of fully-nonlinear multi-directional waves by a 3d viscous numerical wave tank*, Proceedings of the Eleventh (2001) International Offshore and Polar Engineering Conference, 2001.
- [Sen11] U. Senturk, *Modeling nonlinear waves in a numerical wave tank with localized meshless RBF method*, Computers & Fluids **44** (2011), no. 1, 221–228.
- [SSO94] Mark Sussman, Peter Smereka, and Stanley Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys. **114** (1994), 146–159.
- [Tan00] K. Tanizawa, *The State of the Art on Numerical Wave Tank*, Proc. of 4th Osaka Colloquium on Seakeeping Performance of Ships, 2000, pp. 95–114.
- [Té69] R. Témam, *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II)*, Archive for Rational Mechanics and Analysis **33** (1969), 377–385, 10.1007/BF00247696.
- [Voi] *Informationen der Firma Voith Hydro zum Projekt LIMPET*, Website, http://www.voithhydro.com/vh_en_aktuelles_moviegallery_wavepower.htm, letzter Zugriff 17.08.2011.
- [WSH05] Y. Watanabe, H. Seaki, and R.J. Hosking, *Three-dimensional vortex structures under breaking waves*, Journal of Fluid Mechanics **545** (2005), 291–328.
- [WYT95] P. Wang, Y. Yao, and M.P. Tulin, *An efficient numerical tank for non-linear water waves, based on the multi-subdomain approach with BEM*, International Journal for Numerical Methods in Fluids **20** (1995), no. 12, 1315–1336.