



**UNIVERSITÀ POLITECNICA DELLE MARCHE**

**FACOLTÀ DI INGEGNERIA**

Corso di Laurea Magistrale in Ingegneria Meccanica

**Eolico offshore:**

**Modellazione numerica di una piattaforma galleggiante**

**Offshore wind energy:**

**Numerical modelling of a floating platform**

**Relatore:**

Prof. Andrea Crivellini

**Tesi di Laurea di:**

Andrea Falcioni

**Anno accademico:**

2022/2023



*A chi c'era, che forse mi osserva con orgoglio,  
A chi c'è, per la forza che mi dà ogni giorno,  
A chi ancora non c'è, che in fondo di tutto quanto è il senso.*



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Eolico offshore flottante . . . . .	1
1.2	Struttura della tesi . . . . .	6
<b>2</b>	<b>Background teorico</b>	<b>7</b>
2.1	Teorie del moto ondoso . . . . .	7
2.2	Forze di Morison . . . . .	12
2.3	Catenaria e Taut-mooring . . . . .	14
2.3.1	Equazioni della catenaria . . . . .	14
2.3.2	Sistemi di Taut-mooring . . . . .	17
<b>3</b>	<b>Metodi CFD</b>	<b>19</b>
3.1	Equazioni di governo . . . . .	19
3.1.1	Equazioni di continuità e quantità di moto . . . . .	19
3.1.2	Equazioni per la turbolenza . . . . .	21
3.2	Modellazione numerica delle equazioni di governo . . . . .	24
3.2.1	Schema temporale - modello di <i>Crack-Nicolson</i> . . . . .	24
3.2.2	Schema per il gradiente . . . . .	25
3.2.3	Schemi del secondo ordine e <i>Flux Limiters</i> . . . . .	27
<b>4</b>	<b>OpenFOAM</b>	<b>29</b>
4.1	Introduzione a OpenFOAM . . . . .	29
4.2	Generazione della mesh . . . . .	31

4.2.1	<i>blockMesh</i> . . . . .	31
4.2.2	<i>snappyHexMesh</i> . . . . .	32
4.2.3	Mesh dinamica . . . . .	33
4.2.4	Generazione delle onde . . . . .	35
<b>5</b>	<b>Approccio iniziale alla modellazione</b>	<b>37</b>
5.1	Floating Object . . . . .	37
5.1.1	Definizione della geometria . . . . .	38
5.1.2	Condizioni iniziali . . . . .	40
5.2	DTCHullWave . . . . .	42
5.2.1	Importazione di una geometria ‘.stl’ . . . . .	42
5.2.2	Creazione della vasca ondosa . . . . .	43
<b>6</b>	<b>Modello preliminare 2D</b>	<b>45</b>
6.1	Geometria . . . . .	46
6.2	Condizioni iniziali . . . . .	46
6.3	Risultati . . . . .	49
6.4	Comparazione con dati analitici . . . . .	56
<b>7</b>	<b>Modello finale</b>	<b>61</b>
7.1	Geometria . . . . .	61
7.1.1	Caratteristiche della piattaforma . . . . .	61
7.1.2	Mesh . . . . .	64
7.1.3	Moody . . . . .	67
7.2	Condizioni iniziali . . . . .	70
7.3	Risultati . . . . .	71
<b>8</b>	<b>Conclusioni</b>	<b>85</b>
	<b>Bibliografia</b>	<b>89</b>

# Elenco delle figure

1.1	Svizzera, Bourg Saint Pierre, parco solare galleggiante . . .	2
1.2	Danimarca, parco eolico flottante . . . . .	3
1.3	Tipologie di ancoraggi per turbine eoliche flottanti in acque profonde . . . . .	4
1.4	Test in vasca marittima di un particolare basamento galleggiante in scala ridotta . . . . .	5
2.1	Superficie libera dell'onda [5] . . . . .	8
2.2	Condizione cinematica sulla superficie libera . . . . .	9
2.3	Sistema di ormeggio tipico a catenaria . . . . .	15
2.4	Possibili configurazioni della linea di ormeggio a catenaria [9] . . . . .	16
2.5	Schema delle forze di un ormeggio a catenaria a linea singola [9] . . . . .	17
2.6	Sistema di ormeggio taut-mooring tipico . . . . .	18
3.1	Andamento dei principali <i>Flux Limiters</i> utilizzati nella CFD . . . . .	28
4.1	Principali ambienti del software OpenFOAM . . . . .	30
4.2	Struttura fondamentale di un modello su OpenFOAM . . . . .	31

4.3	Sistema di numerazione di vertici, spigoli e facce su Open-FOAM . . . . .	32
4.4	Fasi di ricostruzione della mesh attorno a un oggetto con <i>snappyHexMesh</i> . . . . .	33
4.5	Mesh dinamica: configurazione iniziale e deformata [11] .	34
4.6	Mesh dinamica: visualizzazione della <i>innerDistance</i> e <i>outerDistance</i> nella configurazione iniziale e deformata . . .	35
5.1	Configurazione iniziale del caso <i>floatingObject</i> . . . . .	42
5.2	Zone di raffinamento della mesh del dominio attorno allo scafo . . . . .	43
5.3	Visualizzazione delle onde del caso <i>DTCHullWave</i> . . . .	44
6.1	Raffinamento della mesh attorno alla cassa . . . . .	47
6.2	Misura della superficie libera, onda con $T=1$ s . . . . .	48
6.3	Misura della superficie libera, onda con $T=1.1$ s . . . . .	48
6.4	Misura della superficie libera, onda con $T=1.3$ s . . . . .	49
6.5	Convenzione sui segni . . . . .	50
6.6	Istantanea della simulazione CFD con mesh dinamica, caso A3 . . . . .	50
6.7	Caso A . . . . .	51
6.8	Caso A1 . . . . .	51
6.9	Caso A2 . . . . .	52
6.10	Caso A3 . . . . .	52
6.11	Caso A4 . . . . .	53
6.12	Caso B1 . . . . .	53
6.13	Caso B2 . . . . .	54
6.14	Caso B3 . . . . .	54
6.15	Caso B4 . . . . .	55



6.16	Casi A-B3 . . . . .	55
6.17	Casi A1-B4 . . . . .	56
6.18	Comparazione caso A1 con dati analitici . . . . .	57
6.19	Comparazione caso A2 con dati analitici . . . . .	58
6.20	Comparazione caso B3 con dati analitici . . . . .	58
6.21	Comparazione caso B4 con dati analitici . . . . .	59
7.1	Barge-Based platform . . . . .	62
7.2	Modello 3D della Barge-Based platform con GstarCAD .	63
7.3	Rappresentazione della mesh del modello finale . . . . .	66
7.4	Ciclo di iterazione con l'utilizzo di Moody . . . . .	67
7.5	Posizionamento taut-moorings . . . . .	70
7.6	Onda 1 anno, direzione 0 gradi . . . . .	71
7.7	Onda 100 anni, direzione 0 gradi . . . . .	72
7.8	Onda 1 anno, direzione 90 gradi . . . . .	73
7.9	Onda 100 anni, direzione 90 gradi . . . . .	74
7.10	Direzione 0 gradi . . . . .	75
7.11	Direzione 90 gradi . . . . .	76
7.12	Comparazione dei 4 casi analizzati . . . . .	77
7.13	Snapshot 1 - Onda 100 anni, direzione 0 gradi . . . . .	78
7.14	Snapshot 2 - Onda 100 anni, direzione 0 gradi . . . . .	78
7.15	Snapshot 3 - Onda 1 anno, direzione 90 gradi . . . . .	79
7.16	Snapshot 4 - Onda 1 anno, direzione 90 gradi . . . . .	79
7.17	Postprocessing - pressione sulla struttura . . . . .	81
7.18	Postprocessing - velocità del fluido . . . . .	82
7.19	Postprocessing - pressione e velocità fluido-struttura . . .	82
7.20	Postprocessing - energia cinetica turbolenta - snap 1 . . .	83
7.21	Postprocessing - energia cinetica turbolenta - snap 2 . . .	83

7.22 Postprocessing - vettori velocità all'interfaccia multifase	
- snap 1 . . . . .	84
7.23 Postprocessing - vettori velocità all'interfaccia multifase	
- snap 2 . . . . .	84

# 1

## Introduzione

### 1.1 Eolico offshore flottante

Negli ultimi anni le strutture flottanti stanno ricoprendo un ruolo sempre più importante nel mondo, in quanto possono essere sfruttate per diverse funzioni, quali impianti Oil&Gas, moli, frangiflutti, impianti fotovoltaici, parchi eolici.

Il settore energetico, in generale, responsabile di oltre il 70% delle emissioni globali [10], si pone tra i prossimi obiettivi quello di ridurre significativamente la produzione di CO<sub>2</sub>, per frenare gli effetti del cambiamento climatico, cercando di invertire la rotta della filiera produttiva di ogni suo settore. Per fare ciò è necessario implementare tecnologie e business model sempre più evoluti, e concentrarsi maggiormente sulle energie rinnovabili a nostra disposizione. Un cambio di prospettiva di questa portata permetterà, oltre che un futuro 100% carbon free, di far fronte a una domanda sempre più ingente di energia elettrica.

Tra le varie tecnologie possibili per lo sfruttamento e la produzione di energie rinnovabili, troviamo gli impianti eolici [6], capaci di convertire

## 1. Introduzione

---

in elettricità una particolare forma di energia, il vento.

Quest'ultimo è una forma indiretta di energia solare. L'energia prodotta dal Sole, infatti, è distribuita in modo irregolare sulla superficie del nostro pianeta, concentrandosi maggiormente nelle zone vicine all'equatore. Lo squilibrio termico derivante da questa condizione si traduce in zone a pressione differente; le regioni ad alta pressione tenderanno a muoversi verso quelle a più bassa pressione, spostando grandi volumi di aria, generando così le correnti ventose, una forma di energia pulita e costante. Questa fonte è facilmente sfruttabile da turbine eoliche, posizionabili sia sulla terraferma, che in mare, appunto.

Le turbine offshore, posizionate sia in acque basse che profonde, devo-



Figura 1.1: Svizzera, Bourg Saint Pierre, parco solare galleggiante

no far fronte a specifiche problematiche, come i carichi marini, onde e correnti, e dei sistemi di fissaggio o ancoraggio, mantenendo la stabilità per quanto più possibile. In questo contesto, le strutture galleggianti sono progettate per acque profonde: a differenza dei sistemi tradizionali, fissati al fondale marino alle massime profondità di 60 metri, quelli flottanti sono ancorati al suolo, e sono capaci di resistere anche alle peggiori condizioni di mare aperto.

In Italia, l'eolico offshore flottante sembra essere l'opzione migliore, in quanto il Mediterraneo presenta un'elevata profondità dei fondali, e una soluzione di questo tipo permetterebbe oltretutto una tutela paesaggi-

## 1. Introduzione

---



Figura 1.2: Danimarca, parco eolico flottante

stica, diversamente dalle strutture onshore. Si è visto e studiato come zone quali Puglia, Calabria e le isole sono particolarmente prestanti dal punto di vista eolico. In queste regioni, l'Associazione Nazionale Energia del Vento (ANEV) ha stimato un potenziale offshore di circa 5GW per il 2040. In questo momento sono in atto diversi progetti di parchi eolici galleggianti, come il Barium bay: 74 turbine eoliche flottanti al largo della costa di Bari, capaci di produrre fino a 3 mila gigawattora all'anno, soddisfacendo così, in via teorica, il fabbisogno di oltre un milione di famiglie. *“Il mare Adriatico offre una combinazione vincente dei fattori che rendono lo sviluppo di progetti eolici galleggianti molto attraente per l'Italia: buona ventosità, grande spazio e opportuni allacci alla rete elettrica nazionale. Trovare una valida soluzione per coprire il fabbisogno di energia elettrica per oltre un milione di famiglie in Italia è per noi motivo di una grande motivazione e determinazione”* - Ingmar Wilhelm, AD di Galileo, piattaforma paneuropea per lo sviluppo di energie rinnovabili, che detiene, assieme a Hope (azienda per la creazione di impianti rinnovabili), la paternità del progetto Barium bay. Esistono diversi tipi

## 1. Introduzione

---



Figura 1.3: Tipologie di ancoraggi per turbine eoliche flottanti in acque profonde

di strutture galleggianti, le quali devono soddisfare delle richieste comuni, come la stabilità della posizione entro certi limiti, che si traduce nel controllo degli spostamenti verticali, orizzontali e rotazionali. La risposta dinamica della turbina alle sollecitazioni marine (vento e onde) è valutata tramite spostamenti e rotazioni rigide del centro di massa, condizionati dagli stessi carichi e dai sistemi di ormeggio. La fase di studio e ottimizzazione degli ancoraggi è necessariamente specifica per ogni turbina, in base al luogo in cui essa è posizionata, dal quale dipendono i carichi ambientali, e può essere effettuata in diversi modi. Un primo metodo consiste nell'utilizzo di modelli in scala, in delle vasche specifiche capaci di generare onde e correnti. Questa metodologia è particolarmente utile per lo studio degli effetti di bordo dati dalla geometria tridimensionale, ma ha delle evidenti limitazioni per quanto riguarda gli effetti di scala, sommati agli elevati costi e tempi sperimentali. Un'altra via consiste nell'utilizzo di simulazioni numeriche, tramite software CFD



## 1. Introduzione

---

(Computational Fluid Dynamic) e/o FSI (Fluid-Structure Interaction).

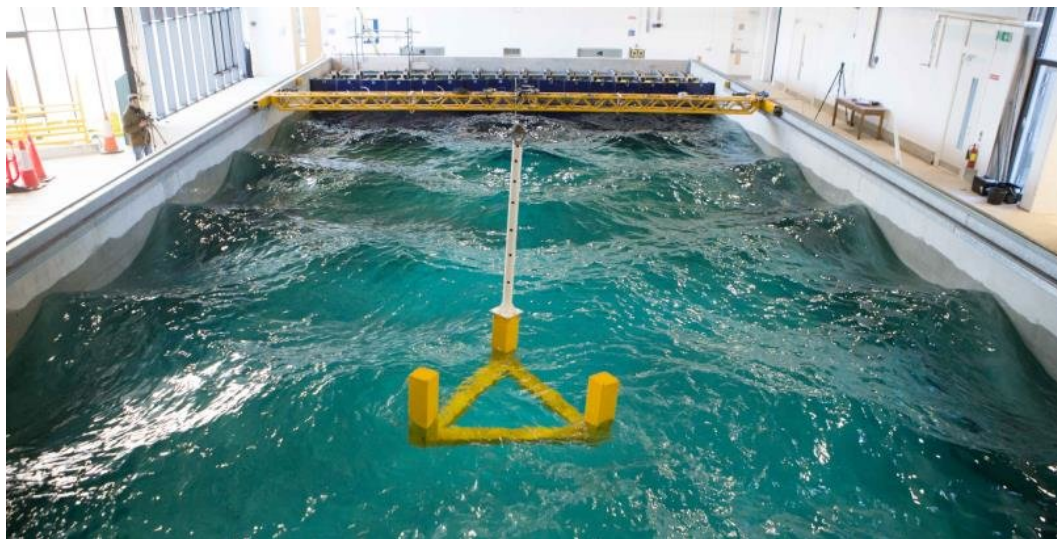


Figura 1.4: Test in vasca marittima di un particolare basamento galleggiante in scala ridotta

Tali modelli devono in ogni caso essere validati da dati sperimentali, così da poter aggiustare i parametri fino ad ottenere i risultati attesi, ottenuti appunto da prove fisiche. Una volta validato, il modello numerico può essere esteso ai casi specifici richiesti, così da ottenere risultati in differenti condizioni di carico o di ormeggio. Il vantaggio dell'utilizzo di modelli numerici, evidentemente, è il minor costo e tempo sperimentale, e la possibilità di effettuare simulazioni in scala naturale. Lo scopo della seguente tesi è quindi quello di realizzare un modello multifase di una Barge-based platform (basamento galleggiante per accogliere una turbina eolica) tramite l'utilizzo di OpenFOAM, per una prima valutazione dell'interazione fluido-struttura e per l'analisi degli spostamenti di quest'ultima sotto l'effetto di forze marittime quali onde e correnti. L'azienda per la quale viene svolto tale progetto è la Techfem S.p.A. collocata nel settore dell'Oil&Gas che si occupa anche di strutture offshore.

### 1.2 Struttura della tesi

La tesi è strutturata come segue.

Nel capitolo introduttivo è stato presentato il tema nel quale il progetto di tesi è inserito: è stata data una visione generale dell'utilizzo dell'eolico offshore, delle richieste di design che comporta, e dei vantaggi che le simulazioni numeriche apportano nella fase progettuale. Nel secondo capitolo verrà riportato il background teorico utile allo studio dell'interazione fluido-struttura, dei moti ondosi e dei sistemi di ancoraggio. Nel terzo capitolo è riportata la teoria che sta dietro alle simulazioni numeriche: partendo dalle equazioni fondamentali che governano la dinamica dei fluidi, ci si concentrerà sui vari modelli numerici che sono implementati nei programmi CFD. Nel quarto capitolo sarà presentato OpenFOAM, il codice CFD utilizzato in questo progetto; verranno spiegate le equazioni di discretizzazione e i solutori utilizzati, facendo riferimento anche al post-processing. Il quinto capitolo è incentrato sulla spiegazione dei primi approcci alla programmazione; verranno illustrati dei primi modelli tutorial, spiegando come ragiona il programma e quali sono i modi per personalizzare un modello base. Nel sesto capitolo verrà presentato il primo modello personalizzato, realizzato in 2D in modo da velocizzare l'apprendimento del software e dei vari parametri di controllo, per arrivare ad una simulazione analiticamente accettabile. Il settimo capitolo parlerà del modello finale 3D realizzato nell'ambito di questo progetto, dove sono stati utilizzati gli esatti ancoraggi e diverse condizioni iniziali, così da ottenere delle prime risposte dal software rispetto al comportamento di una Barge-based platform in acqua. L'ottavo e ultimo capitolo sarà dedicato alle conclusioni e alla puntualizzazione di eventuali progetti futuri collegati.



# 2

## Background teorico

### 2.1 Teorie del moto ondoso

Verrà riportata in seguito la trattazione per descrivere il campo di moto di un'onda nel dominio dello spazio e del tempo, in particolare la teoria del moto ondoso di Airy. Le equazioni che ricaveremo sono di fondamentale importanza, poiché saranno le stesse utilizzate in seguito per generare le condizioni di mare che ci interessano, modificando i parametri secondo le casistiche da sperimentare.

La teoria di Airy [17] è una semplificazione della teoria di Stokes, la quale descrive onde superficiali bidimensionali di ampiezza finita. Le ipotesi che stanno alla base di quest'ultima sono le seguenti [7]:

- i.* Onde regolari: periodiche, progressive, superficiali, bidimensionali;
- ii.* Fluido inviscido, omogeneo, incomprimibile;
- iii.* Campo di moto irrotazionale;
- iv.* Profondità costante e fondo impermeabile;
- v.* Non vi è interazione con altri tipi di moto.

## 2. Background teorico

---

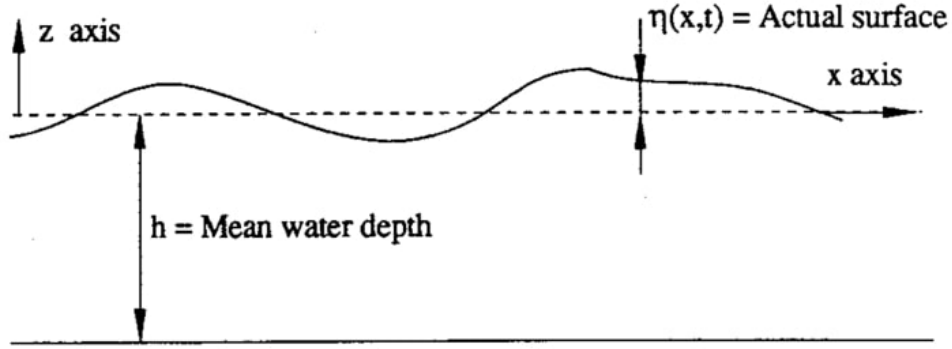


Figura 2.1: Superficie libera dell'onda [5]

Nel caso di Airy, viene aggiunta un'ulteriore ipotesi, al fine di linearizzare le equazioni, e renderle coincidenti con la soluzione di Stokes al primo ordine. L'ipotesi che viene aggiunta è quella di 'onde di piccola ampiezza'. A questo punto andiamo a ricavare le equazioni per descrivere il campo di moto. L'equazione della conservazione della massa per flusso incomprimibile è la seguente:

$$\nabla \cdot \vec{V} = 0 \quad (2.1)$$

Ovvero, considerando che il flusso si sviluppi solamente nel piano XZ,

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad (2.2)$$

Dall'ipotesi di flusso irrotazionale, possiamo determinare le componenti del campo di velocità, una volta definita una funzione scalare del potenziale di velocità  $\psi(x, z, t)$ :

$$\vec{V} = \nabla \psi \quad (2.3)$$

L'equazione di continuità diventa allora

$$\nabla \cdot (\nabla \psi) = \nabla^2 \psi = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} = 0 \quad (2.4)$$

## 2. Background teorico

Quella riportata sopra è l'equazione di Laplace, che è una forma ellittica, e in quanto tale ha bisogno di condizioni al contorno sulla superficie del dominio:

*i.* Condizione cinematica sul fondale,

$$w|_{-h} = \frac{\partial \psi}{\partial z}|_{-h} = 0 \quad \forall x, t \quad (2.5)$$

Anche se a primo impatto sembrerebbe una condizione molto forte, bisogna pensare che il fondale è posto idealmente a distanza infinita, è quindi lecito supporlo impermeabile e perfettamente piatto, poiché la sua geometria risulterà in seguito trascurabile.

*ii.* Condizione cinematica sulla superficie libera,

$$dz = w \cdot dt = \frac{\partial \eta}{\partial t} dt + \frac{\partial \eta}{\partial x} dx \quad (2.6)$$

da cui

$$w = \frac{\partial \psi}{\partial z} = \frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} \quad (2.7)$$

Lo spostamento verticale della particella è dovuto al contribu-

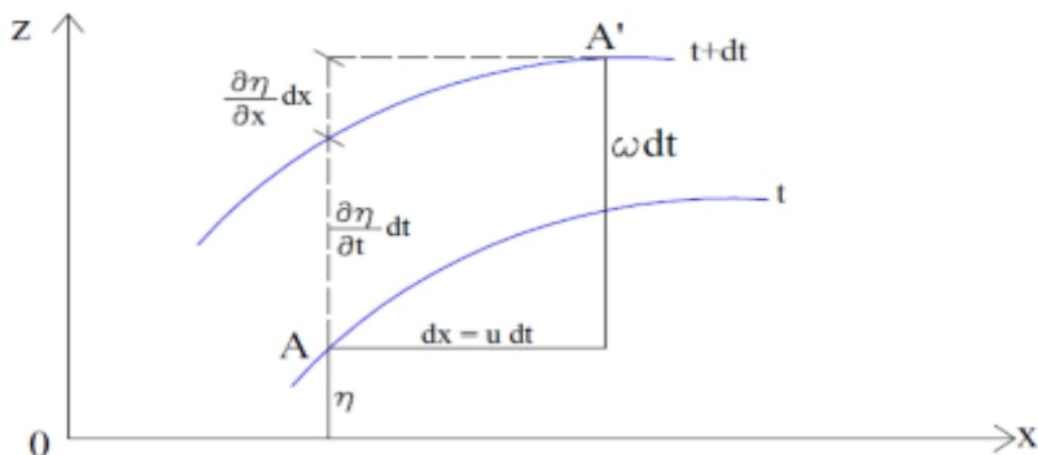


Figura 2.2: Condizione cinematica sulla superficie libera

to dell'innalzamento della superficie libera e dalla traslazione di quest'ultima.

## 2. Background teorico

---

iii. Condizione dinamica sulla superficie libera,

$$\frac{p}{\rho} + \frac{\partial\psi}{\partial t} + \frac{1}{2}(u^2 + w^2) + gz = \frac{p_{atm}}{\rho} \quad (2.8)$$

Sostituendo nella precedente equazione (Bernoulli) la pressione che si ha sulla superficie libera, si ottiene

$$\frac{\partial\psi}{\partial t} + \frac{1}{2}(u^2 + w^2) + gz = 0 \quad (2.9)$$

Il problema può essere quindi riassunto in

$$\left\{ \begin{array}{l} \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial z^2} = 0 \quad \forall x, t, z : -h < z < \eta(x, t), \\ w|_{-h} = \frac{\partial\psi}{\partial z}|_{-h} = 0 \quad \forall x, t, \\ \frac{\partial\eta}{\partial t} + u \frac{\partial\eta}{\partial x} = w \quad \forall x, t : z = \eta(x, t), \\ \frac{\partial\psi}{\partial t} + \frac{1}{2}(u^2 + w^2) + gz = 0 \quad \forall x, t : z = \eta(x, t). \end{array} \right. \quad (2.10)$$

Il problema sopra riportato è ancora troppo complesso per essere risolto analiticamente; utilizziamo allora l'ipotesi di Airy, e supponiamo che l'ampiezza delle onde sia piccola rispetto alla loro lunghezza d'onda (plausibile per onde oceaniche). Si può dimostrare che il sistema 2.10 viene linearizzato in questo modo:

$$\left\{ \begin{array}{l} \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial z^2} = 0 \quad \forall x, t, z : -h < z < \eta(x, t), \\ w|_{-h} = \frac{\partial\psi}{\partial z}|_{-h} = 0 \quad \forall x, t, \\ \frac{\partial\eta}{\partial t} = w \quad \forall x, t, \\ \frac{\partial\psi}{\partial t} = -g\eta \quad \forall x, t. \end{array} \right. \quad (2.11)$$

Si vuole ora verificare che la seguente equazione è soluzione del problema:

$$\psi(x, z, t) = A(z)sen(\omega t - kx + \varphi) \quad (2.12)$$

## 2. Background teorico

---

Calcolando i termini interni all'equazione di Laplace e sostituendo all'interno della prima equazione del sistema 2.11, avremo

$$\left(-k^2 A(z) + \frac{d^2 A(z)}{dz^2}\right) = 0, \forall z \in [-h, \eta] \quad (2.13)$$

Che ha come soluzione

$$A(z) = c_1 \cosh(kz + c_2). \quad (2.14)$$

Sostituendo ora l'equazione esplicita di  $\psi(x, z, t)$  nella seconda equazione del sistema 2.11 otteniamo:

$$kc_1 \sin(\omega t - kx + \varphi) \sinh(-kh + c_2) = 0, \forall x, t \quad (2.15)$$

La quale, per  $c_2 = kh$ , soddisfa i primi due vincoli.

Utilizzando le ultime due equazioni del sistema 2.11, possiamo ora ricavare  $\omega$ :

$$\begin{cases} w = \frac{\partial \eta}{\partial t} = \frac{\omega^2 c_1}{g} \sin(\omega t - kx + \varphi) \cosh(kh) \\ w = \frac{\partial \psi}{\partial z} = kc_1 \sin(\omega t - kx + \varphi) \sinh(kh) \end{cases} \quad (2.16)$$

Che fornisce l'equazione di dispersione

$$\omega^2 = gk \cdot \tanh(kh). \quad (2.17)$$

Da questa relazione capiamo che la pulsazione dell'onda, e di conseguenza il periodo ( $T = 2\pi/\omega$ ), sono funzioni solamente della lunghezza d'onda ( $k = 2\pi/L$ ) e della profondità del fondale [4], e non viene intaccata in nessun modo da eventuali correnti  $U$ . Onde e correnti saranno quindi due input indipendenti, che influenzeranno il periodo reale di ritorno dell'onda,

$$T_r = \frac{2\pi}{\omega + Uk}. \quad (2.18)$$

## 2. Background teorico

---

Il profilo d'onda, infine, sarà

$$\eta(x, t) = a \cdot \cos(\omega_r t - kx) \quad (2.19)$$

dove

$$\omega_r = \frac{2\pi}{T_r}. \quad (2.20)$$

I parametri che andremo a modificare per ricreare le condizioni di mare desiderate saranno quindi  $a, T, U, h$ .

### 2.2 Forze di Morison

La determinazione delle forze che vengono esercitate sulla struttura è di fondamentale importanza per il calcolo di spostamenti e accelerazioni della stessa. È noto dalla letteratura che su un corpo immerso in un fluido reale agisce una forza lungo la direzione del moto  $F_x$ , e una in direzione trasversale  $F_y$ .

$$F_x = \frac{1}{2} \rho C_d A u^2 \quad (2.21)$$

$$F_y = \frac{1}{2} \rho C_l A u^2 \quad (2.22)$$

Queste equazioni descrivono le forze generate da un flusso stazionario. Per stimare i carichi marini (generati da flussi non stazionari) che agiscono sulle strutture offshore viene utilizzata l'equazione di Morison. Tale equazione è semi-empirica, e restituisce la forza esercitata dal fluido sulla struttura nella direzione del moto (le forze trasversali al flusso sono da valutare separatamente).

L'equazione di Morison è costituita dalla somma di due componenti: una componente inerziale  $F_I$ , in fase con l'accelerazione del flusso locale, e una componente di resistenza  $F_D$ , proporzionale al quadrato della

## 2. Background teorico

---

velocità del flusso istantaneo. I due coefficienti idrodinamici empirici presenti nell'equazione (quello d'inerzia e quello di resistenza), sono ricavati sperimentalmente, e dipendono in generale dal numero di Keulegan-Carpenter  $Kc = uT/L$ , dal numero di Reynolds  $Re = uL/\nu$  e dalla rugosità superficiale.

In condizioni di corpo fisso in flusso oscillatorio, l'equazione di Morison sarà:

$$F = F_I + F_D = \rho C_m V \dot{u} + \frac{1}{2} \rho C_d A u |u| \quad (2.23)$$

dove

$\sim \dot{u} = \frac{du}{dt}$  è l'accelerazione del flusso;

$\sim V$  è il volume del corpo;

$\sim C_m = 1 + C_a$  è il coefficiente d'inerzia;

$\sim C_a$  è il coefficiente di massa aggiunta;

$\sim A$  è un'area di riferimento (ad es. l'area della sezione trasversale del corpo perpendicolare alla direzione del flusso);

$\sim C_d$  è il coefficiente di resistenza;

Le forze di portanza oscillatorie, perpendicolari alla direzione di flusso, non sono presenti nell'equazione.

Nel caso in cui anche il corpo si muova con velocità  $v(t)$ , l'equazione diventa:

$$F = F_a + F_{id} + F_{tr} = \rho V \dot{u} + \rho C_a V (\dot{u} - \dot{v}) + \frac{1}{2} \rho C_d A (u - v) |u - v| \quad (2.24)$$

dove

$\sim F_a$  è la forza di Froude-Krylov;

## 2. Background teorico

---

$\sim F_{id}$  è la forza di massa idrodinamica;

$\sim F_{tr}$  è la forza di trascinamento;

La componente ortogonale della forza esercitata dal fluido sul corpo viene usualmente espressa con l'equazione 2.22.

### 2.3 Catenaria e Taut-mooring

La caratteristica principale richiesta alle strutture galleggianti che ospitano le turbine eoliche è la stabilità e di conseguenza la capacità di ridurre le oscillazioni del sistema al fine di minimizzare il fenomeno di fatica a cui sono soggette le varie componenti. Per quanto riguarda i sistemi di ormeggio, vengono utilizzate generalmente due soluzioni:

- i.* Catenaria;
- ii.* Taut-mooring (elementi tesi).

#### 2.3.1 Equazioni della catenaria

Per i sistemi di ormeggio a catenaria, la stabilità del sistema è garantita dal peso proprio della stessa [21]. La catenaria, che è solitamente composta da catena e cavo, collegando il galleggiante con l'ancora, si trova parzialmente sospesa in acqua.

È inoltre presente un tratto appoggiato sul fondale marino che riduce le forze verticali agenti sul sistema di ancoraggio. Quando la struttura galleggiante è in equilibrio, gran parte della catenaria giace sul fondale del mare mentre la restante parte è sospesa. Solitamente il tratto orizzontale è tra le 5 e le 20 volte superiore al tratto verticale. Quando la struttura si sposta orizzontalmente dalla sua posizione di equilibrio, il punto A (figura 2.4) si sposta fra A1 e A4, la lunghezza della parte di



## 2. Background teorico

---

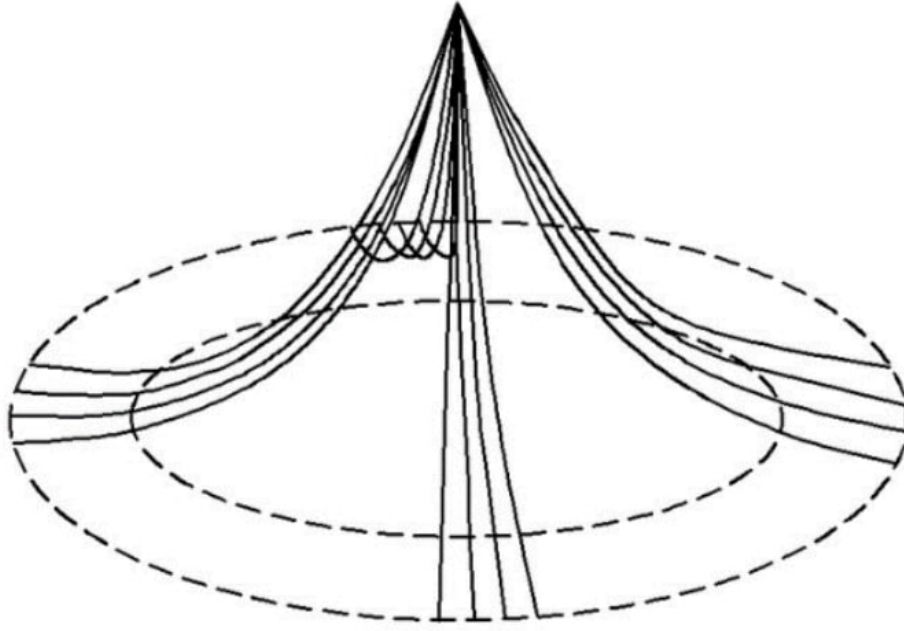


Figura 2.3: Sistema di ormeggio tipico a catenaria

catenaria appoggiata sul fondo si riduce. La tensione è proporzionale alla parte sospesa della catenaria, pertanto il progressivo aumento di linea sospesa comporta un incremento di tensione.

La lunghezza della linea sospesa, trascurando le forze idrodinamiche e considerando il sistema di ormeggio anelastico, si può calcolare come segue:

$$Ls = (v_m + d) \sqrt{\frac{2T_{m,h}}{m(v_m + d)} + 1} \quad (2.25)$$

dove

$\sim v_m$  è la lunghezza verticale della linea sospesa;

$\sim d$  è la distanza tra la cinghia del galleggiante e il pelo libero dell'acqua;

$\sim m$  è il peso unitario della catenaria in acqua;

$\sim T_{m,h}$  è il carico orizzontale applicato alla linea.

## 2. Background teorico

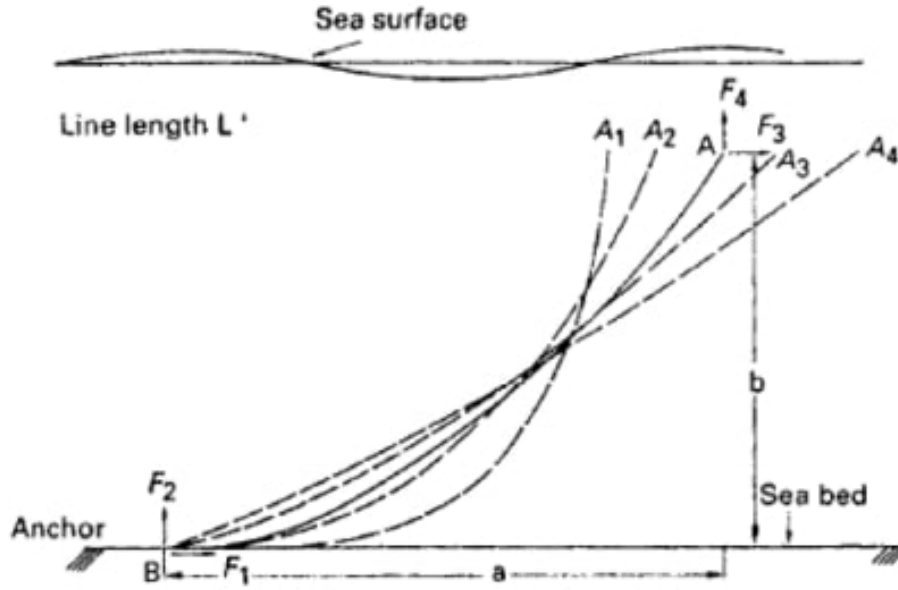


Figura 2.4: Possibili configurazioni della linea di ormeggio a catenaria [9]

La lunghezza orizzontale della linea sospesa può essere ricavata con la seguente formula:

$$x = \frac{T_{m,h}}{m} \cosh^{-1} \left( 1 + \frac{m(v_m + d)}{T_{m,h}} \right) \quad (2.26)$$

mentre l'estensione orizzontale dell'intera linea si calcola come

$$h_m = L_s - (v_m + d) \sqrt{\frac{2T_{m,h}}{mh} + 1} + \frac{T_{m,h}}{m} \cosh^{-1} \left( 1 + \frac{m(v_m + d)}{T_{m,h}} \right) \quad (2.27)$$

Infine, la forza orizzontale si ricava con la seguente equazione:

$$\frac{\partial T_{m,h}}{\partial X} = m \left[ \frac{-2}{\sqrt{\frac{2T_{m,h}}{m(v_m+d)} + 1}} + \cosh^{-1} \left( 1 + \frac{m(v_m + d)}{T_{m,h}} \right) \right] \quad (2.28)$$

## 2. Background teorico

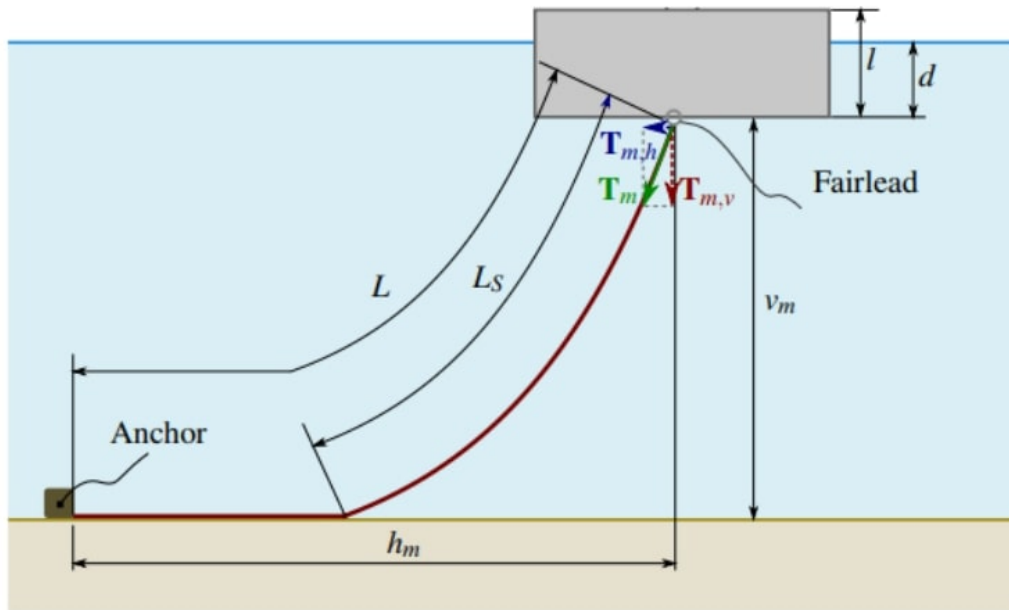


Figura 2.5: Schema delle forze di un ormeggio a catenaria a linea singola [9]

### 2.3.2 Sistemi di Taut-mooring

Quest'altro tipo di ormeggio prevede che la struttura sia collegata al sistema di ancoraggio, posizionato sul fondale marino, tramite elementi tesi. La forza di ripristino, cioè quella utile a riportare il sistema nella posizione di equilibrio, deve essere garantita da un pretensionamento, il quale manterrà inoltre le linee dritte. Per questo tipo di ancoraggi si utilizzano in genere funi metalliche, nylon ad alta resistenza, cavi in poliestere o altri materiali sintetici. I vantaggi di questo sistema rispetto alle catenarie sono diversi:

- i.* In acque profonde, questo tipo di ormeggio riduce largamente la lunghezza e il peso dei cavi; questo si traduce in una minore area occupata dal sistema di ancoraggio, quindi minori rischi di collisione con altri sistemi sottomarini, e un minor costo dei cavi;
- ii.* Le forze di ripristino dell'equilibrio sono più alte, portando a minori spostamenti e rotazioni del sistema galleggiante;

## 2. Background teorico

---

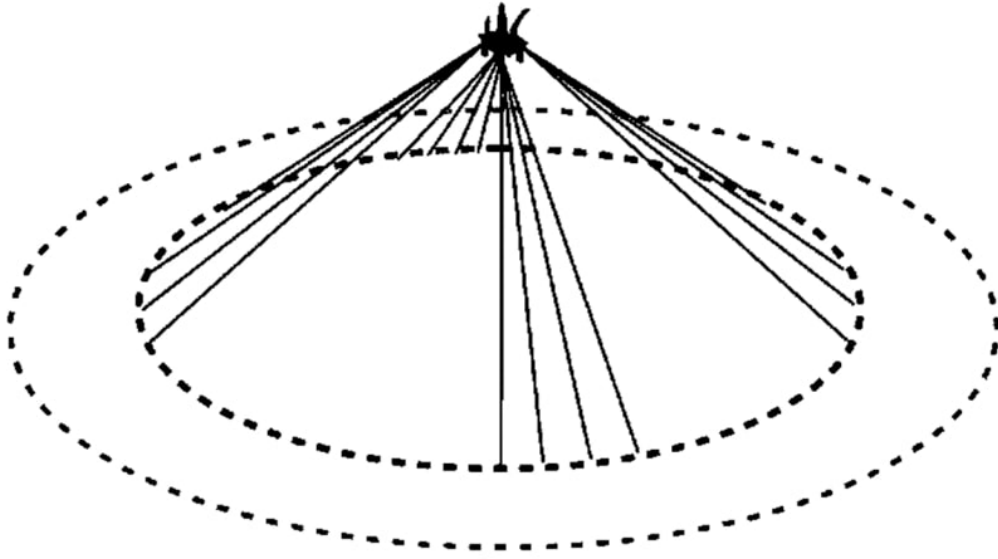


Figura 2.6: Sistema di ormeggio taut-mooring tipico

*iii.* I materiali sintetici come nylon e poliestere hanno delle elevatissime resistenze alla fatica e alla corrosione.

Al contrario, gli svantaggi di questo metodo si traducono in elevati stress sotto dure condizioni marine, elevati costi di fabbricazione e installazione, possibili danneggiamenti dovuti a sedimentazioni infiltratesi all'interno del cavo. Nel presente progetto verrà simulato questo tipo di ancoraggio, tramite la modellazione di un insieme di sistemi massa-molla-smorzatore, dove il coefficiente di smorzamento verrà preso arbitrario (effetto trascurabile), mentre il coefficiente  $k$  della molla sarà uguale a:

$$k = \frac{EA}{L_0} \quad (2.29)$$

dove

$\sim E$  è il modulo di Young del materiale di cui è composto il cavo;

$\sim A$  è l'area della sezione del cavo;

$\sim L_0$  è la lunghezza a riposo del cavo.

# 3

## Metodi CFD

### 3.1 Equazioni di governo

#### 3.1.1 Equazioni di continuità e quantità di moto

Il metodo ai volumi finiti, quello utilizzato per la discretizzazione del modello, sfrutta le equazioni di governo della fluidodinamica espresse nella loro forma conservativa, risolvendole per ogni volume di controllo. L'utilizzo della forma conservativa permette di preservare la conservazione dei flussi all'interno di ogni volume di controllo. Viene assunta inoltre l'ipotesi di flussi incomprimibili, cosicché le equazioni di governo della massa e della quantità di moto possano essere scritte nella seguente forma:

$$\nabla \vec{u} = 0 \quad (3.1)$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{\nabla p}{\rho} + \vec{g} + \nu \nabla^2 \vec{u} \quad (3.2)$$

dove

$\vec{u}$  è il vettore velocità;

### 3. Metodi CFD

---

$\sim \nu = \frac{\mu}{\rho}$  è la viscosità cinematica del fluido (costante);

$\sim \rho$  è la densità del fluido;

Essendo il flusso incomprimibile, l'equazione di continuità non contiene la pressione; per risolvere il campo di pressione, tale equazione viene modificata, applicando l'operatore divergenza alla (3.1.1.2), arrivando alla seguente formulazione:

$$\frac{\nabla^2 p}{\rho} + \nabla \cdot (\vec{u} \cdot \nabla \vec{u}) = 0 \quad (3.3)$$

Quella sopra riportata è l'equazione di Poisson, ellittica per la pressione. In realtà l'equazione di conservazione della quantità di moto viene modificata per la presenza di più fasi fluide (acqua e aria) nel modello, e per la natura turbolenta dei moti simulati,

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla (\rho \vec{u} \vec{u}) - \nabla (\mu_{eff} \nabla \vec{u}) = -\nabla p^* - \vec{g} X \nabla \rho + \nabla \vec{u} \nabla \mu_{eff} + \sigma k \nabla \alpha \quad (3.4)$$

dove

$\sim \mu_{eff} = \mu + \rho \nu_t$  è la viscosità totale effettiva, somma della viscosità molecolare e turbolenta;

$\sim \nu_t$  è la viscosità turbolenta;

$\sim p^*$  è la pressione dinamica;

$\sim X$  è il vettore posizione;

$\sim \sigma$  è il coefficiente di tensione superficiale;

$\sim k$  è la curvatura all'interfaccia multifase;

$\sim \alpha$  è l'indicatore di fase.

### 3. Metodi CFD

---

Quest'ultimo indicatore, compreso tra 0 e 1, è ciò che identifica la fase del fluido in cui il volume di controllo si trova:  $\alpha = 0$  indica che la fase del volume è aria,  $\alpha = 1$  significa che ci troviamo in fase acquosa;  $\alpha = 0.5$  indica che il volume è costituito da parti uguali di acqua e aria. Le proprietà intensive sono espresse interpolando linearmente, in funzione di  $\alpha$ , i valori appartenenti ad ogni fase; ad esempio per la densità avremo

$$\rho = \alpha \rho_f + (1 - \alpha) \rho_b \quad (3.5)$$

L'equazione di trasporto per  $\alpha$  è

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{u} \alpha) = 0 \quad (3.6)$$

Vedremo in seguito come vengono utilizzate queste equazioni dai software numerici, in particolare quali sono gli schemi di discretizzazione implementati in OpenFOAM.

#### 3.1.2 Equazioni per la turbolenza

Esistono vari approcci per la modellazione della turbolenza, come la DNS (Direct Numerical Simulation) che risolve numericamente le equazioni di Navier-Stokes alle scale di Kolmogorov più piccole [3], o la LES (Large Eddy Simulation) che risolve solamente le strutture vorticosi più grandi, modellando poi quelle più piccole. Nell'ambito di questo progetto verrà utilizzato il metodo RANS (Reynolds Averaged Navier-Stokes), o delle equazioni mediate alla Reynolds: in questo modello ci si riferisce non più alle grandezze istantanee, ma a quelle mediate in un certo periodo di tempo, sufficientemente piccolo rispetto al fenomeno da analizzare, sufficientemente grande rispetto alle scale di tempo più piccole

### 3. Metodi CFD

---

della turbolenza. Il moto turbolento viene così descritto come sovrapposizione di un moto medio nel tempo più una fluttuazione istantanea della grandezza d'interesse. Ad esempio, per la velocità abbiamo

$$u_i = \bar{u}_i + u'_i \quad (3.7)$$

$\sim \bar{u}_i$  termine mediato;

$\sim u'_i$  termine fluttuante;

Mediando nel tempo le equazioni di governo istantanee si ottengono così le equazioni mediate alla Reynolds:

$$\nabla \cdot \bar{u} = 0 \quad (3.8)$$

$$\nabla \cdot (\bar{u} \cdot \bar{u}) = -\frac{\nabla \bar{p}}{\rho} + \nabla \cdot [\bar{\tau}] - \nabla \cdot \overline{u'u'} \quad (3.9)$$

dove  $\overline{u'u'} = [\tau^R]$  è il tensore degli sforzi di Reynolds, modellato come segue (ipotesi di Boussinesq)

$$[\tau^R] = 2\nu_t [\bar{S}] - \frac{2}{3} [I] K \quad (3.10)$$

$\sim [\bar{S}]$  è il tensore medio delle deformazioni;

$\sim K$  è l'energia cinetica turbolenta.

Una volta definite le RANS, la turbolenza può essere modellata in diversi modi: modelli a zero equazioni come le wall-fuctions, modelli a un'equazione come lo Spallart-Almaras, modelli a due o più equazioni come il  $K - \epsilon$  o il  $K - \omega$ . Nell'ambito di questa tesi viene utilizzato il modello  $K - \omega$ ; questa metodologia risolve la turbolenza grazie all'aggiunta di due equazioni, una per l'energia cinetica turbolenta  $K$ , e una per la frequenza caratteristica delle scale più piccole,  $\omega$ . Viene qui di seguito



### 3. Metodi CFD

---

riportato il sistema risolutivo finale con le due equazioni aggiuntive senza la dimostrazione matematica, per non appesantire ulteriormente la trattazione.

$$\left\{ \begin{array}{l} \nabla \cdot \bar{u} = 0 \\ \frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\bar{u} \cdot \bar{u}) = -\frac{\nabla p^*}{\rho} + \nabla \cdot (2(\nu + \nu_t) [\bar{S}]) \\ \frac{\partial K}{\partial t} + \bar{u} \cdot \nabla K = \nu_t \bar{S}^2 + \nabla \cdot \left( \left( \nu + \frac{\nu_t}{\sigma_K} \right) \nabla K \right) - \varepsilon \\ \frac{\partial \omega}{\partial t} + \nabla \cdot (\bar{u} \omega) = (C_{1\varepsilon}) \nu_t \bar{S}^2 \frac{\omega}{K} - (C_{2\varepsilon} - 1) \omega^2 + \nabla \cdot \left( \left( \nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \right) \\ \quad + 2 \frac{\nu_t}{\sigma_\omega K} \nabla \omega \cdot \nabla K \end{array} \right. \quad (3.11)$$

Analizzando il secondo membro delle equazioni per la  $K$  e per la  $\omega$  si evidenzia un termine di produzione di energia turbolenta, uno per il trasporto/diffusione e uno per la dissipazione della stessa. Nell'ultima equazione, inoltre, è presente un termine aggiuntivo che permette di risolvere le equazioni nel *viscous sub-layer* (sotto-strato laminare), dove le fluttuazioni di velocità si annullano a favore degli sforzi viscosi. Questi tre termini sono in accordo con la teoria della 'cascata di energia' di Richardson, che spiega come l'energia prodotta alle scale più grandi si trasferisca in maniera non viscosa fino alle scale più piccole, dove la viscosità turbolenta dissipa totalmente l'energia, eliminando il vortice; senza il termine di produzione, infatti, non sarebbe spiegata la costanza dell'energia cinetica turbolenta nel tempo, essendo presente, in quel caso, solamente una dissipazione viscosa.

## 3.2 Modellazione numerica delle equazioni di governo

Le soluzioni analitiche delle equazioni alle derivate parziali sono espressioni in forma chiusa che descrivono la variazione della variabile dipendente in maniera continua in funzione delle variabili indipendenti. Al contrario, le soluzioni numeriche forniscono informazioni solo in alcuni punti, chiamati punti di griglia. Per far sì che il computer possa risolvere numericamente le equazioni originali, quindi in modo approssimato, queste vengono discretizzate e trasformate in equazioni algebriche, dove gli operatori differenziali vengono sostituiti da relazioni che si basano sullo sviluppo in serie di Taylor. Esistono diversi metodi di discretizzazione; verranno spiegati qui di seguito alcuni modelli discretizzati utilizzati poi per il progetto di tesi.

### 3.2.1 Schema temporale - modello di *Crack-Nicolson*

Viene ora spiegata la strategia generale dei metodi alle differenze finite per risolvere il problema della discretizzazione temporale, generalmente descritto come

$$\begin{cases} \frac{\partial u}{\partial t} = f(u, t) \\ u(t = t_0) = u_0 \end{cases} \quad (3.12)$$

Si divide l'intervallo d'integrazione  $I = [t_0, t_f]$  in  $N$  sotto-intervalli  $I_k = [t_k, t_{k+1}]$   $n = 0, 1, \dots, N - 1$ , di ampiezza uniforme  $\Delta t = (t_f - t_0)/N$ . Vogliamo determinare i valori dell'incognita  $u$  agli istanti  $t_1, t_2, \dots$  a partire dalla conoscenza del suo valore all'istante  $t_0$  :  $u_1 = u(t_1), u_2 = u(t_2), \dots, u_k = u(t_k)$ . La successione dei valori  $u_1, u_2, \dots, u_k, \dots$  costituisce la soluzione del problema 3.12. Bisogna quindi determinare

### 3. Metodi CFD

---

un'approssimazione dell'operatore di derivata temporale in termini di sole operazioni algebriche (somme, differenze, ...). I metodi di discretizzazione sono tanti, e differiscono tra loro per varie caratteristiche, come l'ordine di accuratezza, la stabilità o la consistenza. Il modello utilizzato nell'ambito di questo progetto è il metodo di *Crack-Nicolson*. Tale modello è una discretizzazione del secondo ordine, incondizionatamente stabile, espressa come segue:

$$u_{k+1} = u_k + \frac{\Delta t}{2} [f(u_{k+1}, t_{k+1}) + f(u_k, t_k)] \quad (3.13)$$

Il metodo di Crack-Nicolson fornisce, in forma implicita,  $u_{(k+1)}$  in funzione di  $u_k$ . Ad ogni iterazione bisogna risolvere un'equazione algebrica non lineare. Il vantaggio di questo metodo è la sua stabilità incondizionata, che permette di non avere restrizioni sul valore del  $\Delta t$  utilizzato. Un metodo stabile, infatti, è tale se, ad ogni errore introdotto, quest'ultimo non viene amplificato dallo schema stesso. Questa caratteristica, insieme alla consistenza, ci garantisce la convergenza dello schema (legge di Lax); ciò vuol dire che, all'infittirsi della discretizzazione ( $\Delta t, \Delta x \rightarrow 0$ ), la soluzione numerica tende alla soluzione analitica delle equazioni alle derivate parziali d'origine.

#### 3.2.2 Schema per il gradiente

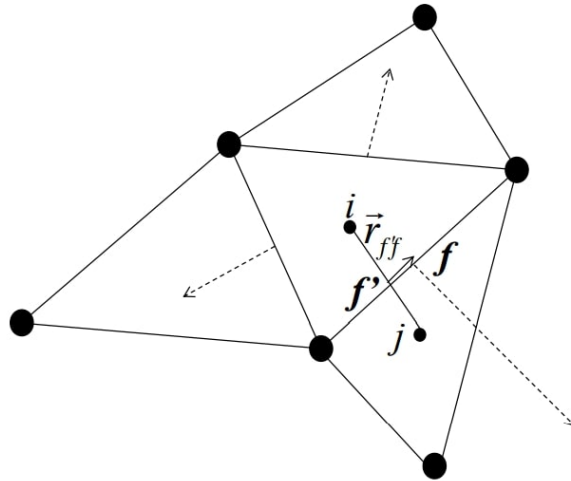
Nella tecnica ai volumi finiti, per giungere a una soluzione, devono essere definiti i flussi all'interfaccia tra due volumi. Ad esempio, per un caso monodimensionale, partendo ad esempio dalla forma conservativa dell'equazione di convezione per la  $u$ , avremo:

$$\int_{\Omega} \frac{\partial u}{\partial t} d\Omega = - \int_{\Omega} \frac{\partial au}{\partial x} d\Omega = - \oint_{\partial\Omega} (au) n d\sigma = - \left( au_{i+\frac{1}{2}} - au_{i-\frac{1}{2}} \right) \quad (3.14)$$

### 3. Metodi CFD

---

Il valore del flusso deve essere calcolato esattamente al centro della faccia, così che l'integrale del flusso sul contorno del dominio sia esattamente uguale al valore ricostruito per l'area del bordo (ipotizzando che il flusso sia una funzione lineare). Se la griglia è irregolare, non basta interpolare la soluzione tra due baricentri per ottenere la ricostruzione esattamente al centro della faccia tra i due volumi.



Bisognerà effettuare una correzione della soluzione del tipo (esempio per la  $u$ )

$$u_f = u_{f'} + \nabla u_{f'} \cdot \vec{r}_{f'f}. \quad (3.15)$$

Definendo

$$\sim u_{f'} = g u_{(i)} + (1 - g) u_{(j)};$$

$$\sim g = \frac{d_{j f'}}{d_{i f'}} = \frac{\text{distanza tra } j \text{ e } f'}{\text{distanza tra } i \text{ e } f'};$$

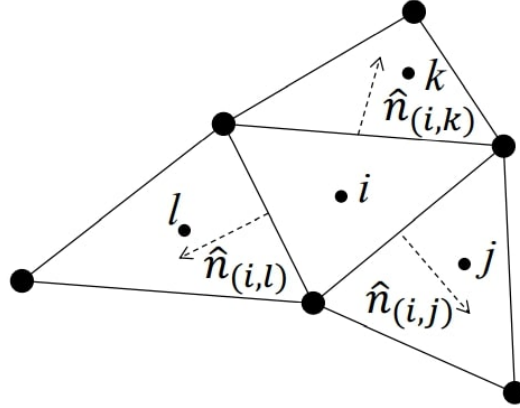
$$\sim \nabla u_{f'} = g \nabla u_{(i)} + (1 - g) \nabla u_{(j)};$$

Possiamo riscrivere la 3.15 come segue

$$u_f = g \left( u_{(i)} + \vec{r}_{f'f} \cdot \nabla u_{(i)} \right) + (1 - g) \left( u_{(j)} + \vec{r}_{f'f} \cdot \nabla u_{(j)} \right) \quad (3.16)$$

### 3. Metodi CFD

Per ottenere una soluzione del secondo ordine necessito quindi di una ricostruzione del gradiente all'interfaccia tra i due volumi.



Anche in questo caso sono disponibili varie tipologie di ricostruzione del gradiente; in questo progetto verrà utilizzata una ricostruzione *Cell-Based* basata sul teorema di Gauss:

$$\int_{\Omega} \nabla u d\Omega = - \oint_{\partial\Omega} u \hat{n} d\sigma \cong \sum_{m=i,j,k} u_{f(i,m)} S_{(i,m)} \hat{n}_{(i,m)} \quad (3.17)$$

da cui possiamo ricavare

$$\nabla u_{(i)} \cong \frac{1}{\Omega_i} \sum_{m=i,j,k} [g(u_{(i)} + \vec{r}'_{f'f} \cdot \nabla u_{(i)}) + (1-g)(u_m + \vec{r}'_{f'f} \cdot \nabla u_{(m)}) S_{(i,m)} \hat{n}_{(i,m)}] \quad (3.18)$$

Essendo  $\nabla u_{(i)}$  presente ai due lati dell'equazione, si utilizza una procedura iterativa, per arrivare poi a una soluzione del valore di faccia 'corretta'.

#### 3.2.3 Schemi del secondo ordine e *Flux Limiters*

Per gli schemi del secondo ordine, il valore della soluzione ricostruito sull'interfaccia tra due volumi, può essere scritto come quello del primo

### 3. Metodi CFD

ordine, più una correzione che dipende dal tipo di interpolazione eseguita sulla faccia (tipo di gradiente utilizzato).

$$f_{i+\frac{1}{2}} = f_i + \psi(r)(f_{i+1} - f_i) \quad (3.19)$$

dove  $r = \frac{f_i - f_{i-1}}{f_{i+1} - f_i}$ .

Essendo gli schemi del secondo ordine affetti da oscillazioni in prossimità di discontinuità o forti gradienti della soluzione, in alcuni casi, si vuole che questi siano monotoni; ciò vuol dire che non devono essere generati nuovi massimi o minimi locali, e che quelli già presenti non devono essere amplificati (*monotonicity preserving*). Si utilizzano quindi delle funzioni  $\psi(r)$  dette *Flux Limiters*, che limitano la variazione del gradiente, così da preservare la monotonia della soluzione.

Nel presente trattato gli schemi del secondo ordine utilizzano la formulazione UpWind del secondo ordine, con una correzione esplicita basata sul valore del gradiente locale di cella.

Viene inoltre utilizzata, nella discretizzazione di  $\alpha$ , la funzione di Van Leer, una *Flux Limiter* di questo tipo:

$$\psi(r) = \frac{r + |r|}{1 + r} \quad (3.20)$$

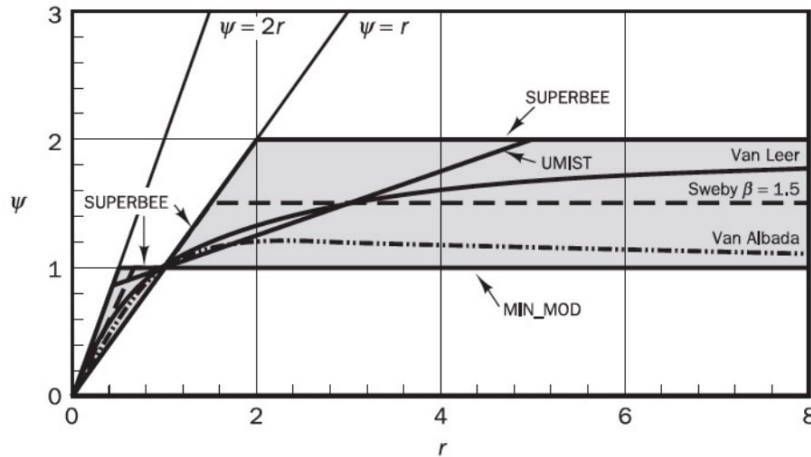


Figura 3.1: Andamento dei principali *Flux Limiters* utilizzati nella CFD

# 4

## OpenFOAM

### 4.1 Introduzione a OpenFOAM

OpenFOAM (Open Field Operation and Manipulation) è un software open source CFD per la risoluzione di sistemi di equazioni alle derivate parziali tramite il metodo dei Volumi Finiti, su griglie poliedriche arbitrarie [2], che utilizza pacchetti di funzioni contenuti in circa 100 librerie in C++. OpenFOAM è dotato in partenza di circa 250 applicazioni che ricadono in due macrosistemi:

- ~ *solvers*, progettati per risolvere specifici problemi nell'ambito della fluidodinamica (reazioni chimiche, turbolenza, trasmissione di calore, ecc. . . ) o della meccanica del continuo;
- ~ *utilities*, pensati per eseguire compiti che coinvolgono manipolazione di dati.

Questa collezione di applicazioni può essere ampliata e modificata in base alle proprie esigenze, così da costruire un personale modello che risolva un problema specificatamente progettato. OpenFOAM è fornito

## 4. OpenFOAM

---

di ambienti pre- e post-processing, i quali anch'essi fanno parte delle *utilities* del programma.

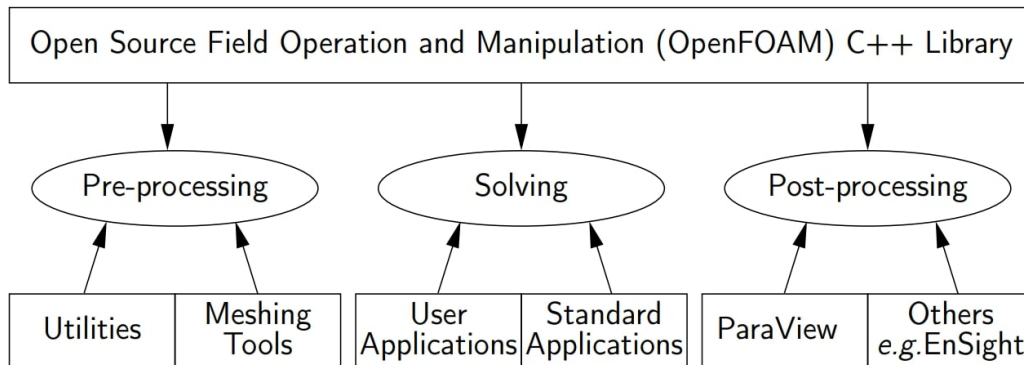


Figura 4.1: Principali ambienti del software OpenFOAM

Ogni caso sviluppato in OpenFOAM ha la propria cartella, strutturata come in figura 4.2. La cartella *system* contiene almeno 3 file fondamentali:

- ~ *fvSolution* che descrive quale tipo di solutore viene utilizzato;
- ~ *fvSchemes* per la configurazione degli schemi di discretizzazione;
- ~ *controlDict* per il controllo della simulazione, i timestep e l'immagazzinamento dei dati.

Nella cartella *constant* sono situati diversi files necessari per la definizione delle proprietà di base del modello, come l'accelerazione gravitazionale, la turbolenza, i modelli di trasporto. In *constant* è presente, inoltre, la sottocartella *polyMesh*, che contiene tutti i file per la descrizione della geometria del modello e della mesh (*points*, *faces*, *owner*, *neighbour*, *boundary*). Nelle *time directories* sono presenti i dati immagazzinati per ogni intervallo di tempo scelto in *controlDict*. Prima di iniziare la simulazione sarà necessario creare una cartella *0*, per descrivere le *boundary condition* iniziali.



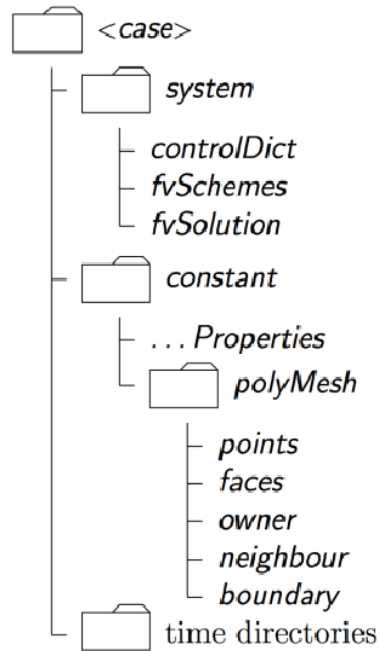


Figura 4.2: Struttura fondamentale di un modello su OpenFOAM

## 4.2 Generazione della mesh

### 4.2.1 *blockMesh*

OpenFOAM opera sempre in un sistema di riferimento cartesiano a tre dimensioni, e tutte le geometrie generate sono tridimensionali. OpenFoam risolve quindi tutti i casi in tre dimensioni di default, ma può essere specificatamente richiesto di risolvere un caso bidimensionale, operando sulle condizioni al contorno del dominio. Lo strumento, presente in OpenFOAM, *blockMesh*, genera una mesh strutturata a blocchi nel dominio computazionale, partendo da una descrizione specifica di esso, posta all'interno del file *blockMeshDict*, presente nella cartella *system*. Da questo file è possibile controllare ogni parametro della mesh, come il numero di suddivisioni di un lato o la grandezza del volume della cella, l'introduzione di fattori di *bias* per il raffinamento di zone specifiche o la descrizione delle *named selections*.

Con *blockMesh*, tramite la dictionary *blockMeshDict*, è possibile anche

## 4. OpenFOAM

---

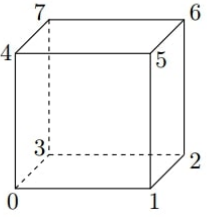
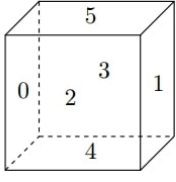
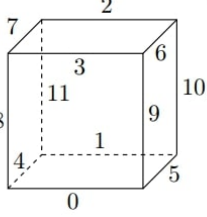
Cell type	Keyword	Vertex numbering	Face numbering	Edge numbering
Hexahedron	hex			

Figura 4.3: Sistema di numerazione di vertici, spigoli e facce su OpenFOAM

generare blocchi di forma più complessa, ma nell'ambito di questa tesi, essendo stato utilizzato solamente per la generazione di una mesh esaedrica nel dominio computazionale, non verranno descritte ulteriori funzioni di questo strumento.

### 4.2.2 *snappyHexMesh*

Lo strumento *snappyHexMesh* [13] genera delle mesh tridimensionali formate da esaedri o esaedri spaccati, partendo da una geometria in formato *Stereolithography* (STL) o *Wavefront Object* (OBJ).

Prima di eseguire *snappyHexMesh* è necessario creare una mesh esaedrica in background, con un aspect ratio di circa 1, e che intercetti con almeno un lato la geometria da includere. Questo passaggio può essere eseguito con *blockMesh* (4.2.1). Il solutore a questo punto inizia a modellare le celle attorno alla superficie .stl, raffinando la zona di contatto tra quest'ultima e il dominio computazionale, rimuovendo infine tutte le celle all'interno dell'oggetto.

*snappyHexMesh* incorpora diversi parametri (che non verranno esposti nell'ambito di questa tesi) per assicurare una buona qualità della mesh.

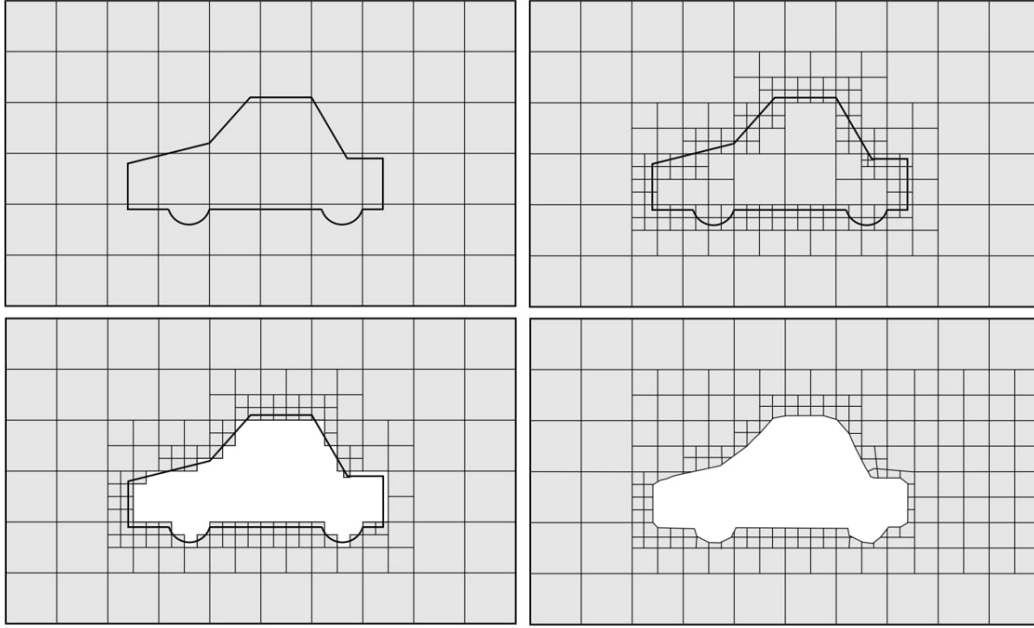


Figura 4.4: Fasi di ricostruzione della mesh attorno a un oggetto con *snappyHexMesh*

### 4.2.3 Mesh dinamica

L'interazione fluido-struttura è gestita tramite l'utilizzo dello strumento *dynamicMotionSolverFvMesh*, nel file *dynamicMeshDict* [8] [12]. Grazie a questo tool è possibile modificare la mesh nel tempo attorno a dei *boundaries* specifici (ad esempio attorno ad un oggetto che si muove con spostamenti rigidi). Il solutore calcola lo spostamento della griglia in base alla pressione calcolata in questi *boundaries*, e fornisce un feedback della loro velocità locale, oltre che della posizione e l'orientamento del centro di massa dell'oggetto in movimento.

Il vantaggio di una mesh dinamica, piuttosto che un remeshing, è naturalmente la riduzione dei tempi di calcolo, mantenendo invariata la qualità della griglia attorno all'oggetto in movimento.

Mantenere la qualità invariata vuol dire garantire la positività dei volumi e delle facce delle celle, la convessità di queste ultime, conservare parametri come *aspect ratio*, *non-orthogonality* e *skewness* entro i range specificati. È possibile inoltre effettuare un raffinamento della zona de-

## 4. OpenFOAM

---

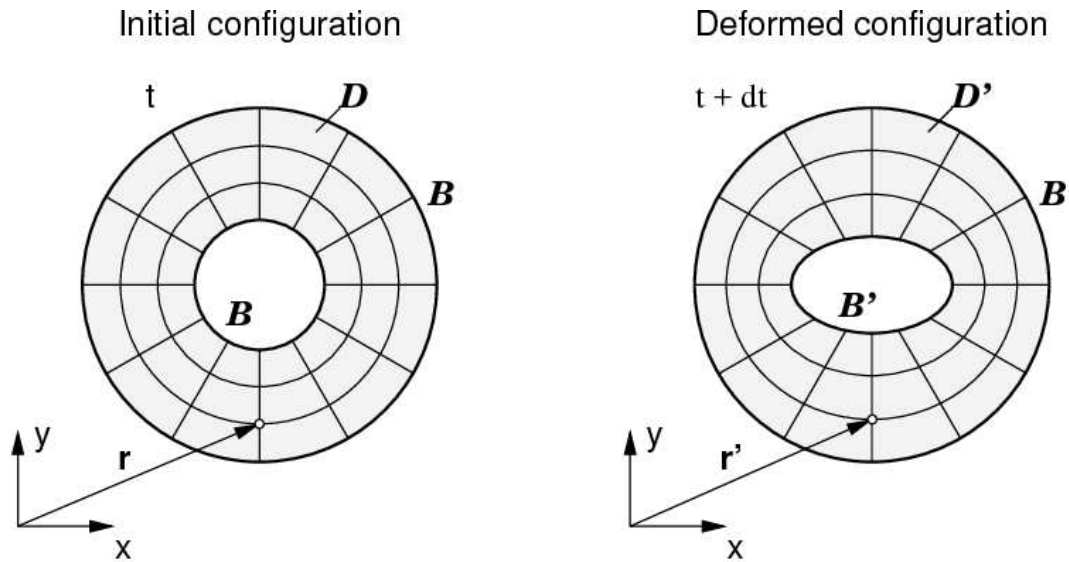


Figura 4.5: Mesh dinamica: configurazione iniziale e deformata [11]

formabile, piuttosto che creare una mesh fine in tutto il dominio statico, così da ridurre i tempi computazionali. In un dominio multifase, con un oggetto galleggiante, ad esempio, è buona norma raffinare la mesh nell'interfaccia tra i due fluidi e attorno all'oggetto, lasciando il resto del dominio più grossolano, in modo tale da fornire comunque risultati affidabili.

Specificando i parametri *innerDistance* e *outerDistance* si controllano le zone deformabili della mesh: tutti i nodi all'interno della *innerDistance* si muovono rigidamente con l'oggetto in movimento; tutti i nodi all'esterno della *outerDistance* rimangono fissi nella posizione iniziale; i nodi posizionati tra la *innerDistance* e la *outerDistance* vengono deformati, così da garantire la continuità e la validità della griglia.

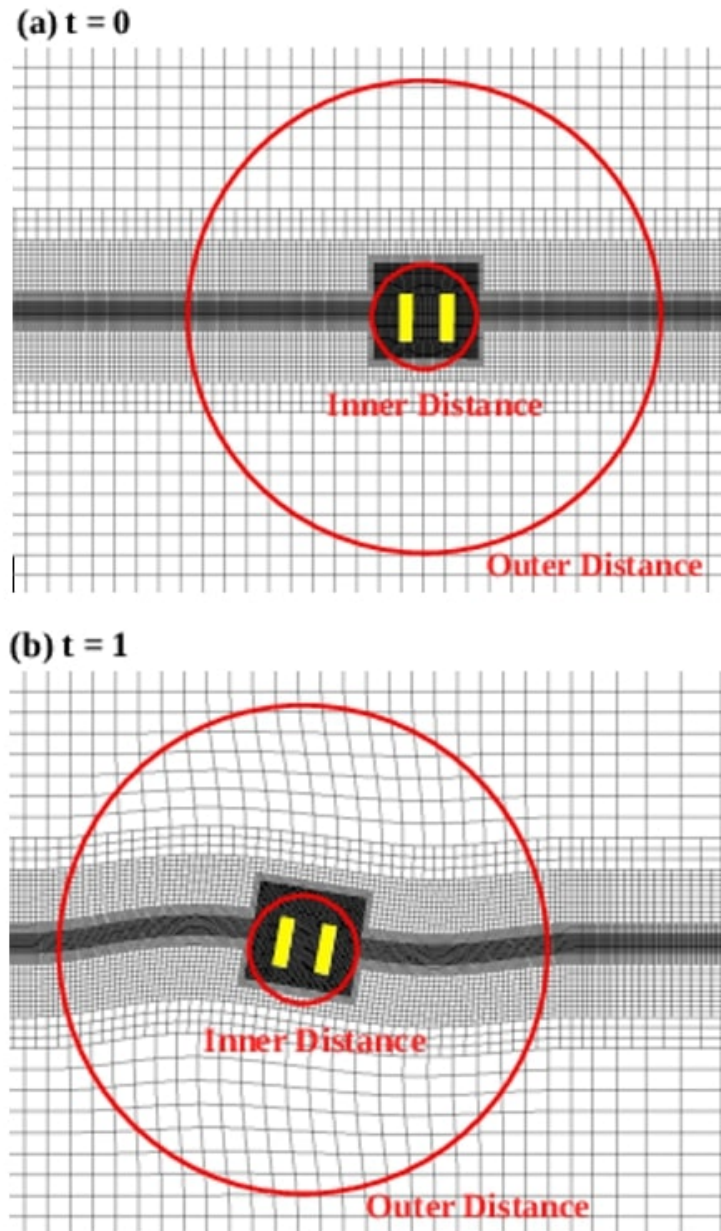


Figura 4.6: Mesh dinamica: visualizzazione della *innerDistance* e *outerDistance* nella configurazione iniziale e deformata

### 4.2.4 Generazione delle onde

OpenFOAM contiene anche modelli per la generazione di onde [19], per descrivere quindi l'altezza dell'onda e la velocità del fluido in una data locazione. In questa tesi verrà usato il modello di Airy (2.1), ma sono presenti anche i modelli di Stokes del primo, secondo e quinto ordine. Per il flusso in ingresso sarà necessario imporre due condizioni al

## 4. OpenFOAM

---

contorno, *waveVelocity* e *waveAlpha*. Con lo strumento *setWaves* sarà quindi possibile applicare il modello d'onda in tutto il dominio per l'inizializzazione del caso. Questo strumento è in grado di eseguire una mappatura accurata al secondo ordine tra il modello d'onda analitico e i volumi finiti. L'assorbimento dell'onda è eseguito imponendo all'uscita una condizione *inletOutlet* e una velocità media del flusso a quella in entrata, così che venga rispettata la conservazione della massa. Altre funzioni possono essere implementate con l'installazione di ulteriori librerie, ambito che non riguarda questa tesi. In seguito (5.2.2) verrà mostrato in dettaglio come impostare le condizioni in *inlet* e *outlet* per ottenere l'onda desiderata.

# 5

## Approccio iniziale alla modellazione

Verranno presentati in questo capitolo dei casi di base presenti in OpenFOAM. L'introduzione di questi casi è di fondamentale importanza, poiché questi verranno utilizzati in seguito per costruire un modello personalizzato completamente nuovo. Da ogni caso verrà estrapolato un particolare strumento o una specifica funzione, e verranno questi poi uniti e adattati secondo le esigenze del presente progetto.

### 5.1 Floating Object

Il caso in esame simula il galleggiamento di un parallelepipedo posto in acqua. Vedremo quindi quali sono gli strumenti presenti in OpenFOAM per risolvere l'interazione multifase fluido-struttura, qual è la formulazione corretta per la generazione di una mesh dinamica [20], e come costruire un oggetto direttamente in OpenFOAM svincolandolo dai gradi di vincolo. Queste operazioni verranno poi estrapolate e personalizzate in analogia con il caso che dovremo modellare.

## 5. Approccio iniziale alla modellazione

---

### 5.1.1 Definizione della geometria

Per prima cosa, nella cartella *system* andiamo a creare il file *blockMeshDict*, per generare la mesh dell'intero dominio computazionale. In questo file saranno descritti i vertici, i lati, i blocchi e le *named selections*. Tramite quest'ultimo passaggio daremo un nome alle facce che rappresentano il contorno del dominio computazionale o degli oggetti presenti in esso, così da andare in seguito ad assegnargli uno specifico tipo di *boundaries*. In questo caso avremo i muri (fondale compreso), l'atmosfera e l'oggetto galleggiante.

Si nota che gli *stationaryWalls* e il *floatingObject* sono descritti come *wall*; questo perché voglio imporre in queste facce una condizione no-slip (flusso nullo e formazione dello strato limite). D'altra parte, l'atmosfera è definita come *patch*: in questo modo il flusso attraverso questa superficie non è necessariamente nullo, ma entrante o uscente, in base a ciò che succede nel dominio computazionale.

La posizione e la geometria del *floatingObject* viene descritta in un file specifico posizionato anch'esso nella cartella *system*, chiamato *topoSetDict*. Essendo l'oggetto galleggiante, e quindi non fisso, la mesh da generare deve essere dinamica; devo quindi creare un file nominato *dynamicMeshDict* nella cartella *constant*, che descrive come deve muoversi l'oggetto e la mesh (solidale ad esso). Utilizzando il solutore *sixDoFRigidBodyMotion* [18] posso descrivere massa, momento d'inerzia e centro di massa dell'oggetto, così come la *innerDistance* e la *outerDistance* della mesh; in questo modo OpenFOAM calcolerà i movimenti rigidi del corpo nei 6 gradi di libertà, in base all'interazione dell'oggetto con il fluido, modificando di volta in volta la mesh attorno ad esso.



## 5. Approccio iniziale alla modellazione

---

```
boundary
(
  stationaryWalls    // Descrizione facce laterali e fondo
  {
    type    wall;    // Flussi nulli attraverso le facce
    faces
    (
      (0 3 2 1)    // Descrizione delle facce tramite i vertici
      (2 6 5 1)    // in senso antiorario rispetto alla
      (1 5 4 0)    // direzione uscente (i vertici sono stati
      (3 7 6 2)    // indicati in precedenza)
      (0 4 7 3)
    );
  }
  atmosphere        // Descrizione della faccia superiore
  {
    type    patch;  // Flussi tramite le facce calcolati
    faces
    (
      (4 5 6 7)
    );
  }
  floatingObject    // Descrizione dell'oggetto galleggiante
  {
    type    wall;
    faces    (); // Descrizione nel file topoSetDict
  }
);
```

Listing 5.1: Estratto di codice da *blockMeshDict*

## 5. Approccio iniziale alla modellazione

---

È inoltre possibile bloccare alcuni gradi di libertà, applicando dei *constraints* al corpo, come un punto fisso (*fixedPoint*) un movimento lungo un asse (*fixedAxis*) o una rotazione rigida attorno ad un asse (*fixedLine*) [1]. Un'ulteriore funzione disponibile permette di applicare al corpo delle forze esterne, tramite dei *restraints* [15]. Il solutore calcolerà così lo spostamento e la rotazione rigida del centro di massa sommando il contributo dato dalle forze di interazione con il fluido e quelle date dai *restraints*, bloccando i movimenti lungo i *constraints* applicati.

### 5.1.2 Condizioni iniziali

Come già esposto nel capitolo 4.1, le condizioni iniziali e la definizione delle variabili nei boundaries vanno sotto una cartella nominata  $\theta$ , posizionata nella cartella generale del caso in esame. Qui avrò tutti i file utili per l'inizializzazione della simulazione. In questo specifico caso avrò i seguenti files:

- ~  $k$  definisce l'energia cinetica turbolenta;
- ~  $\epsilon$  descrive l'energia di dissipazione turbolenta;
- ~  $\nu_t$  descrive la viscosità;
- ~  $p\text{-}rgh$  per la pressione;
- ~  $pointDisplacement$  per gli spostamenti dei boundaries (fissi, imposti o calcolati);
- ~  $\alpha\text{Water}$  per definire il campo  $\alpha$  nei boundaries;
- ~  $U$  definisce la velocità dei flussi nei punti appartenenti ai boundaries.

Per descrivere il campo  $\alpha$  iniziale nell'intero dominio, sarà necessario creare un file nominato *setFieldsDict* nella cartella *system*, per poi uti-

## 5. Approccio iniziale alla modellazione

---

lizzare lo strumento *setFields*. In questo caso il pelo libero dell'acqua è impostato a  $0.5368\text{ m}$  sull'asse  $Z$ , e avremo uno 'scalino' d'acqua iniziale di altezza  $0.75\text{ m}$ ; all'inizio della simulazione lo scalino inizierà a cadere perturbando l'acqua, che farà oscillare l'oggetto galleggiante, il quale è obbligato ad oscillare lungo l'asse  $Y$  passante per il punto  $(0.5\ 0.45\ 0.1)$ . non vi è ancora la presenza di onde o correnti, e il contorno del dominio è un *wall*, ciò vuol dire che le perturbazioni verranno riflesse dalle pareti stesse.

```
defaultFieldValues // Valore di alpha di default
(
    volScalarFieldValue    alpha.water    0
);
regions // Descrizione delle regioni del dominio con alpha=1
(
    boxToCell // Altezza pelo libero dell'acqua
    {
        box (-100 -100 -100) (100 100 0.5368);
        fieldValues ( volScalarFieldValue alpha.water 1 );
    }
    boxToCell // "Scalino" d'acqua
    {
        box (0.5 0.8 -100) (100 100 0.75);
        fieldValues ( volScalarFieldValue alpha.water 1 );
    }
);
```

Listing 5.2: Estratto di codice da *setFieldsDict*

## 5. Approccio iniziale alla modellazione

---

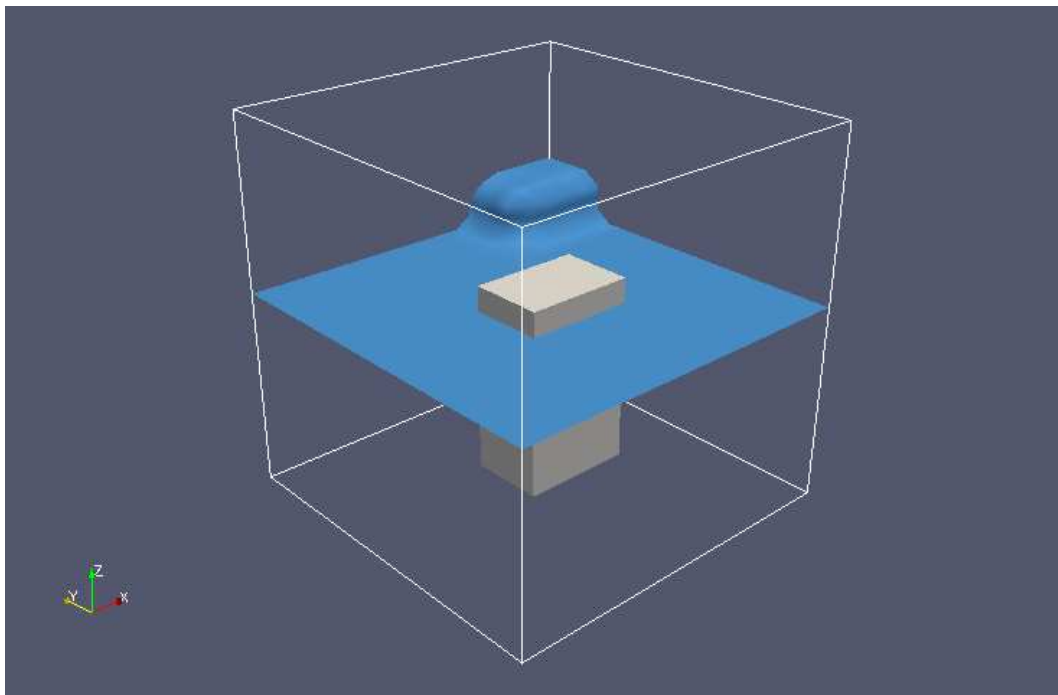


Figura 5.1: Configurazione iniziale del caso *floatingObject*

## 5.2 DTCHullWave

In questo secondo caso, viene simulato il comportamento di uno scafo posto in acqua, in presenza di onde e correnti. L'intera geometria dello scafo è importata in OpenFOAM in formato 'stl'; verrà spiegato quindi come importare una qualsiasi geometria da un software CAD esterno, e come utilizzare con successo lo strumento *snappyHexMesh*. Sarà approfondito quindi il concetto di generazione di un'onda, le condizioni al contorno necessarie e come modificare i parametri specifici a piacimento.

### 5.2.1 Importazione di una geometria 'stl'

Il modello dello scafo è contenuto nel file *Hull.stl*, nella cartella *constant/triSurface*. Il formato *Stereolithography* approssima le superfici di un modello solido con dei triangoli, e non porta con sé informazioni aggiuntive aldilà della posizione di vertici e facce dell'oggetto. Per la

## 5. Approccio iniziale alla modellazione

---

lettura della geometria è sufficiente eseguire il comando *surfaceFeatureExtract*. Se il file *.stl* non è danneggiato (non sono presenti superfici libere o ‘buchi’ nell’oggetto), il comando *snappyHexMesh* creerà effettivamente l’oggetto all’interno del nostro dominio, raffinando e splittando le celle vicine alla superficie, e rimuovendo quelle interne. Con dei parametri specifici è possibile controllare la zona e il livello di raffinamento della mesh, così come la zona di rimozione (potrei voler rimuovere tutte le celle al di fuori dell’oggetto, lasciando solamente quelle interne a quest’ultimo).

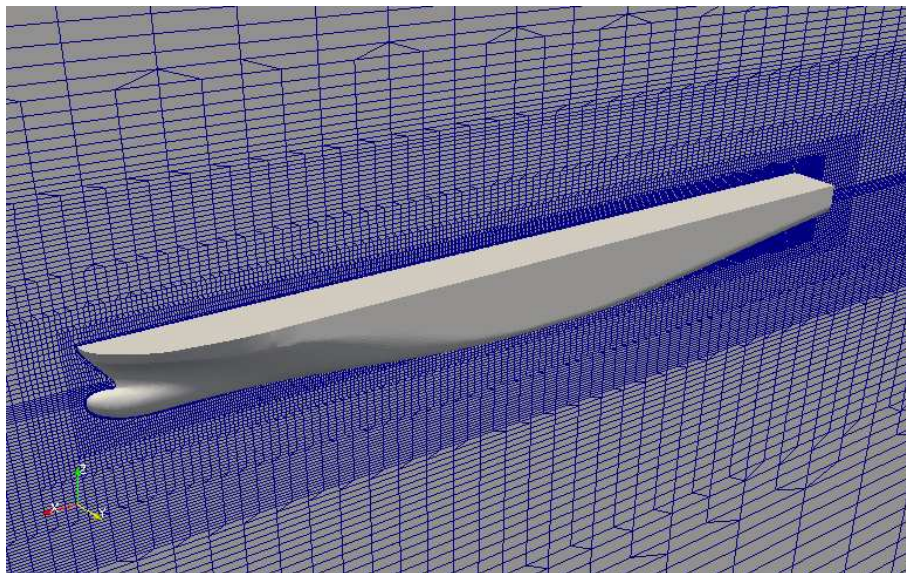


Figura 5.2: Zone di raffinamento della mesh del dominio attorno allo scafo

### 5.2.2 Creazione della vasca ondososa

Nel caso in esame, il dominio computazionale attorno allo scafo viene impostato in modo tale da generare onde regolari a corrente non nulla. Il settaggio di tale configurazione viene eseguito modificando i files *U* e *alpha.water* all’interno della cartella *0* come segue [16]:

~ In *alpha.water*

- L’inlet è impostato come *waveAlpha*;

## 5. Approccio iniziale alla modellazione

---

- L'outlet è del tipo *variableHeightFlowRate*;
- L'atmosfera è definita come *inletOutlet*;
- Lo scafo è descritto come *zeroGradient*;

~ In  $U$

- L'inlet è *waveVelocity*. Viene qui descritto anche il tipo di onda (modello teorico, origine, direzione, velocità, ampiezza, lunghezza, fase, angolo iniziale);
- L'outlet settato come *outletPhaseMeanVelocity*;
- L'atmosfera è del tipo *pressureInletOutletVelocity*;
- Lo scafo è un *movingWallVelocity*.

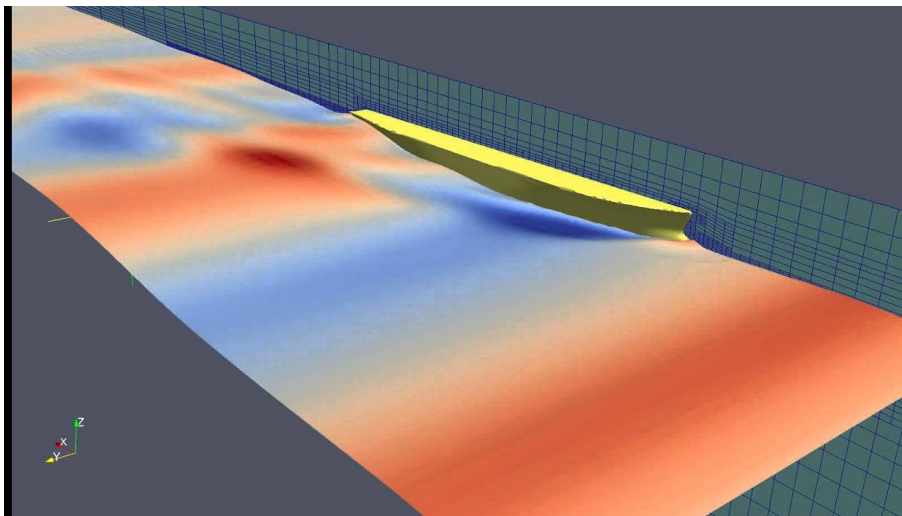


Figura 5.3: Visualizzazione delle onde del caso *DTCHullWave*

Con il comando *scale* è inoltre possibile scalare l'ampiezza dell'onda; questa funzione verrà utilizzata per smorzare l'onda a valle dello scafo (scalando l'ampiezza dal valore 1 al valore 0), così da avere un'uscita del flusso dal dominio più pulita e controllata.

# 6

## Modello preliminare 2D

Al fine di poter creare un modello capace di simulare con buona precisione il comportamento di una piattaforma galleggiante che ospiti una turbina eolica, è necessario prendere confidenza con i parametri che governano le equazioni di riferimento, con i solutori e le utilities del software. Per fare ciò, si è deciso di lavorare, almeno in maniera preliminare, su un modello 2D in scala più piccola, in modo da poter effettuare le simulazioni in tempi brevi, potendo sfruttare minori potenze di calcolo e una memoria contenuta, e poter così modificare i parametri più volte fino ad ottenere risultati validi e comprensibili del sistema simulato. Viene mostrato di seguito il modello 2D finale, frutto di numerose modifiche e perfezionamenti alle condizioni al contorno, ai *restraints*, alla definizione dei *boundaries*, ai parametri di governo del moto solido e fluido. Questo modello, infatti, verrà confrontato con dei risultati analitici, ottenendo degli errori generalmente accettabili. Solamente dopo ci si concentrerà sulla costruzione di un modello tridimensionale di una piattaforma eolica (seppure con le dovute semplificazioni), con una geometria più complessa e degli ancoraggi più specifici, potendo essere più

## 6. Modello preliminare 2D

---

sicuri di non sprecare tempo di calcolo in simulazioni che porterebbero a risultati non affidabili.

### 6.1 Geometria

Il modello 2D è costituito da una cassa galleggiante, che ha una lunghezza di  $0.304m$ , un'altezza di  $0.137m$  e una densità di  $660kg/m^3$  (la profondità è considerata unitaria). La cassa è agganciata al suolo da due molle di rigidità  $1000N/m$ , ancorate ai due vertici inferiori del rettangolo con un angolo  $\theta$  di attacco rispetto al suolo di  $90^\circ$  nei casi A e  $60^\circ$  nei casi B; la lunghezza a riposo delle molle è di  $0.47m$ , così da garantire, tramite pretensionamento, che le molle lavorino sempre in trazione. L'acqua ha una profondità di  $0.65m$ , così che il box risulti completamente immerso di circa  $3cm$ . La vasca ondosa ha una lunghezza di  $15m$ , e un'altezza di  $1.2m$ . Il baricentro dell'oggetto, infine, è inizializzato in  $X = 1m$ ,  $Z = 0.55m$ . La mesh generata è generalmente grossolana, e viene raffinata solamente nei pressi della cassa, comprendendo anche la superficie libera dell'acqua.

### 6.2 Condizioni iniziali

La vasca ondosa è stata creata sfruttando gli stessi settaggi utilizzati in 5.2.2. L'elevata lunghezza di quest'ultima rispetto alle dimensioni della cassa è necessaria, nei casi in cui la corrente è nulla, per evitare riflessioni d'onda nei primi 15 secondi di simulazione (l'outlet in questi casi è impostato come *wall*). Le molle sono simulate introducendo due *linearSpring* nel file *dynamicMeshDict*, nella directory *constant*, e hanno  $k = 1000N/m$ ,  $damping = 100Ns/m$ ,  $restLength = 0.47m$ , con un



## 6. Modello preliminare 2D

---

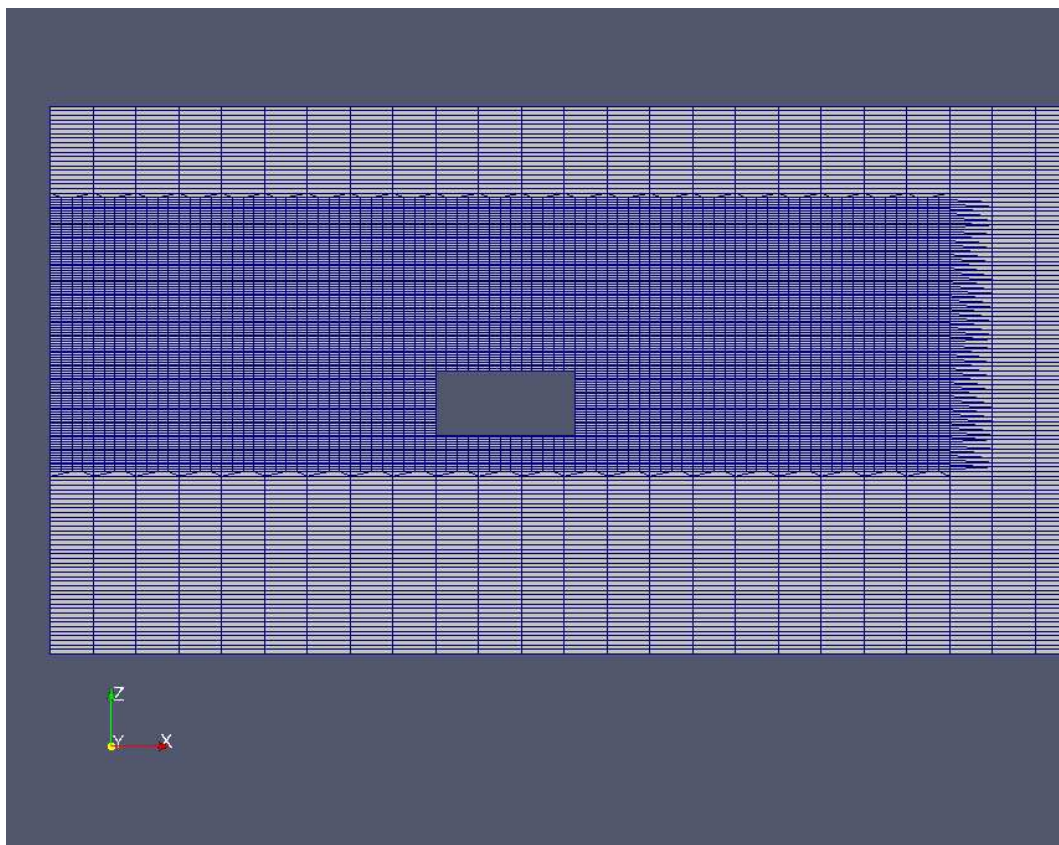


Figura 6.1: Raffinamento della mesh attorno alla cassa

angolo di attacco  $\theta = 90^\circ$  nei casi *A* e  $\theta = 60^\circ$  nei casi *B*. Al fine di verificare la stabilità del modello, sono state simulate diverse condizioni di mare, modificando, caso per caso, velocità della corrente, oltre che ampiezza e periodo dell'onda. Le varie casistiche effettuate sono riportate in tabella 6.1, e sono le stesse utilizzate in [14].

Le diverse condizioni ondose sono state prima simulate in assenza di cassa galleggiante, così da verificare la bontà e la stabilità della vasca marittima.

## 6. Modello preliminare 2D

Caso	U	Ampiezza	Lunghezza	Periodo
	[m/s]	[m]	[m]	[s]
<b>A</b>	-	0.73	2.46	1.3
<b>A1</b>	0.25	-	-	-
<b>A2</b>	0.5	-	-	-
<b>A3</b>	0.25	0.73	2.46	1.15
<b>A4</b>	-	0.45	1.55	1
<b>B1</b>	-	0.08	1.55	1
<b>B2</b>	-	1	2.46	1
<b>B3</b>	-	0.08	2.46	1.3
<b>B4</b>	0.25	-	-	-

Tabella 6.1: Condizioni di moto ondoso, casistiche simulate

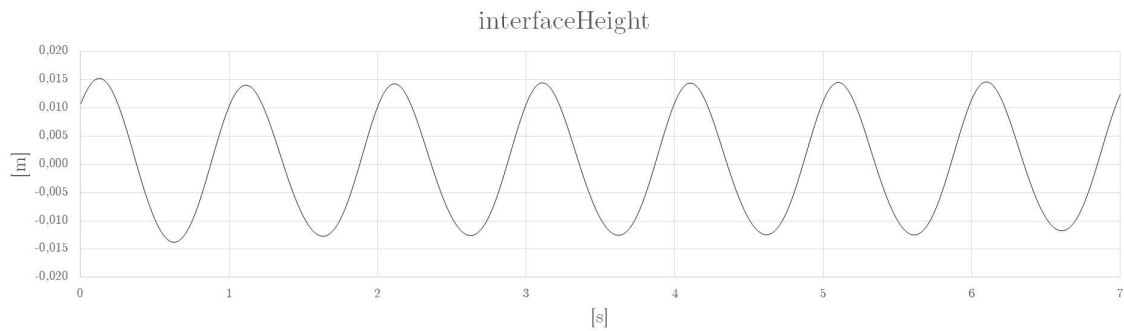


Figura 6.2: Misura della superficie libera, onda con  $T=1$  s

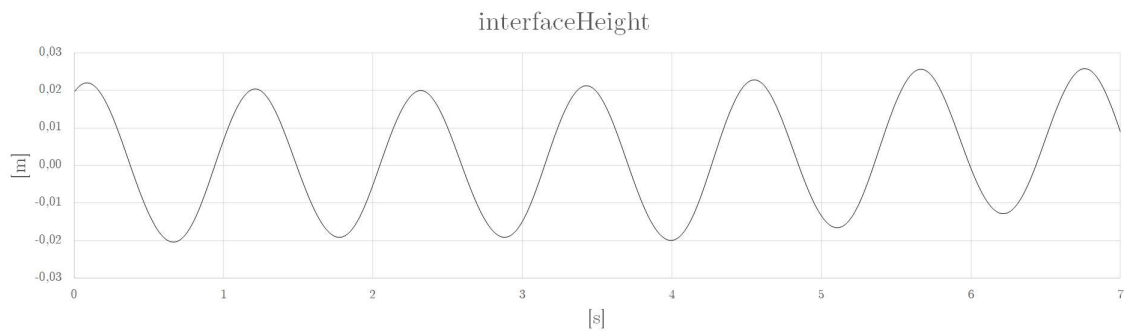


Figura 6.3: Misura della superficie libera, onda con  $T=1.1$  s

## 6. Modello preliminare 2D

---

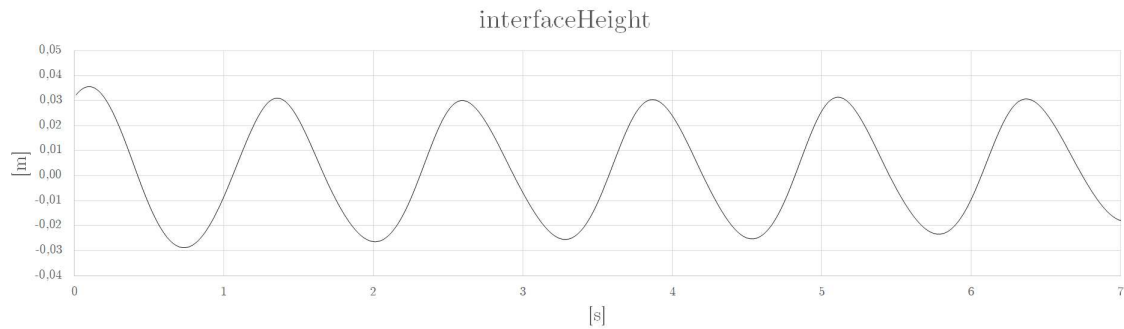


Figura 6.4: Misura della superficie libera, onda con  $T=1.3$  s

### 6.3 Risultati

Di seguito si riportano i risultati delle varie simulazioni effettuate (tramite spostamenti del centro di massa della cassa). Per quanto i dati in uscita possano essere confermati analiticamente, questi risultati non possono essere validati sperimentalmente, in quanto gli ancoraggi realmente utilizzati in ambito offshore non si comportano come molle ideali: questi infatti non esercitano forze in compressione, ma rispondono solamente a sforzi di trazione. L'utilizzo di molle, inoltre, costringe all'impiego di un coefficiente elastico basso, così da permettere alle forze di galleggiamento di mantenere gli ancoraggi in tensione; un coefficiente elastico troppo alto produrrebbe spostamenti troppo bassi, ai quali (a seguito di rotazioni anche minime date dalle condizioni ondose) conseguirebbero stati di trazione-compressione alterni, che porterebbero all'instabilità del sistema, e infine alla divergenza della simulazione. Il comportamento generale di questo modello discosta quindi da quelli sperimentali presenti in letteratura, ma è chiaro che, in caso di validazioni analitiche, l'unica fonte di errore è l'utilizzo, appunto, di ancoraggi diversi. La convenzione sui segni è riportata in figura 6.5, dove *Heave* è il galleggiamento (spostamento del baricentro lungo l'asse verticale), *Surge* è l'ondeggiamento nella direzione di propagazione dell'onda, e *Pitch* è la rotazione del cor-

## 6. Modello preliminare 2D

po attorno all'asse perpendicolare al piano formato dalle direzioni di Heave e Surge; quest'ultima verrà menzionata nei risultati del modello 3D come *Rotation*, essendo l'unica rotazione abilitata.

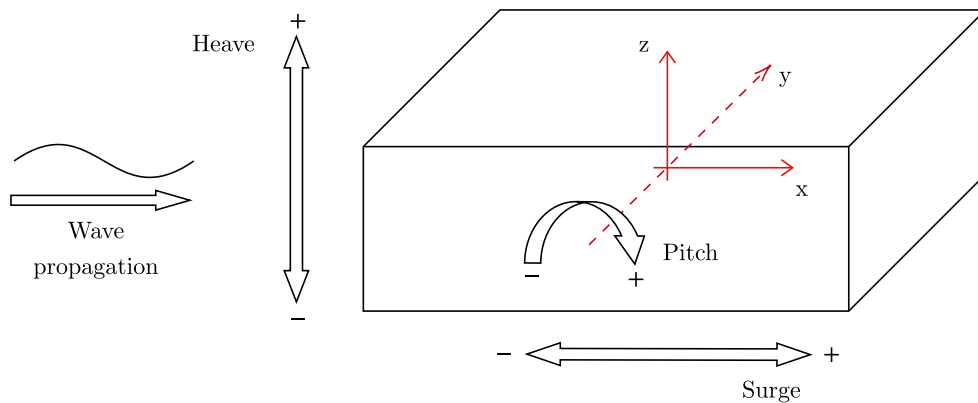


Figura 6.5: Convenzione sui segni

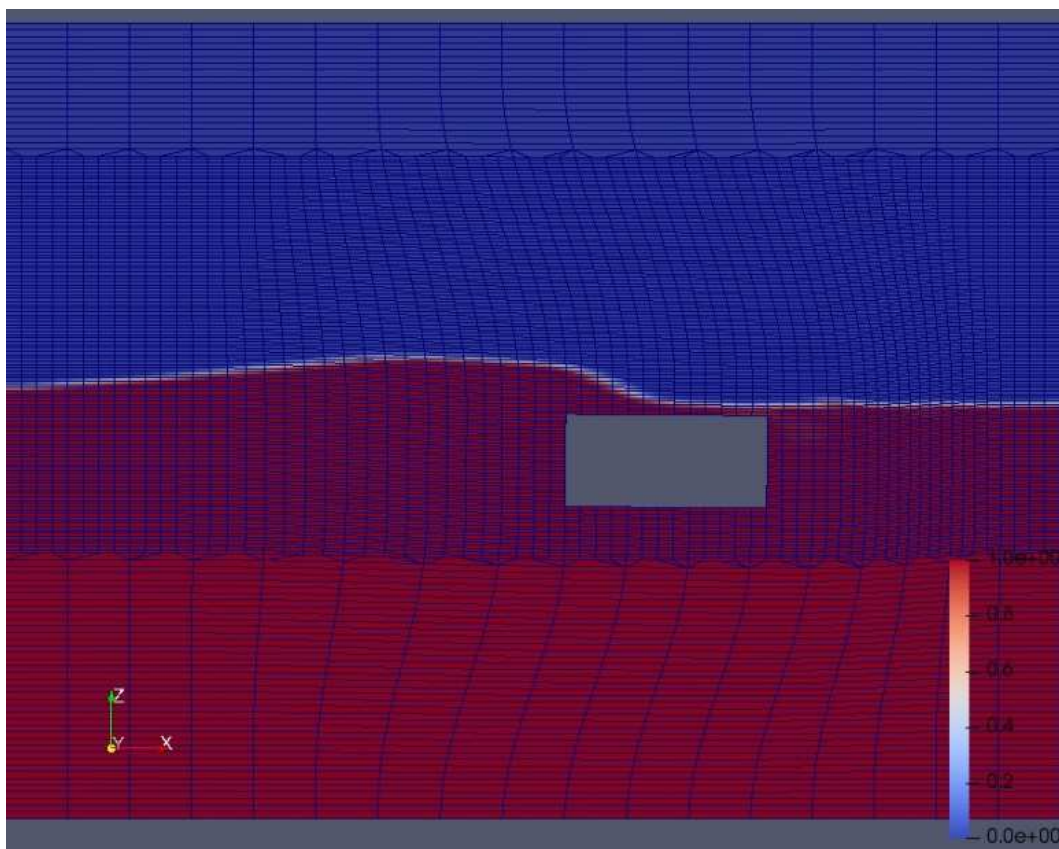


Figura 6.6: Istantanea della simulazione CFD con mesh dinamica, caso A3

## 6. Modello preliminare 2D

---

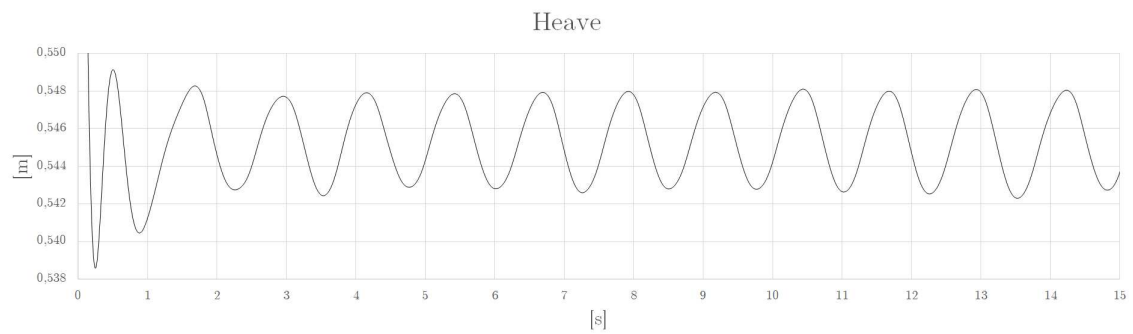
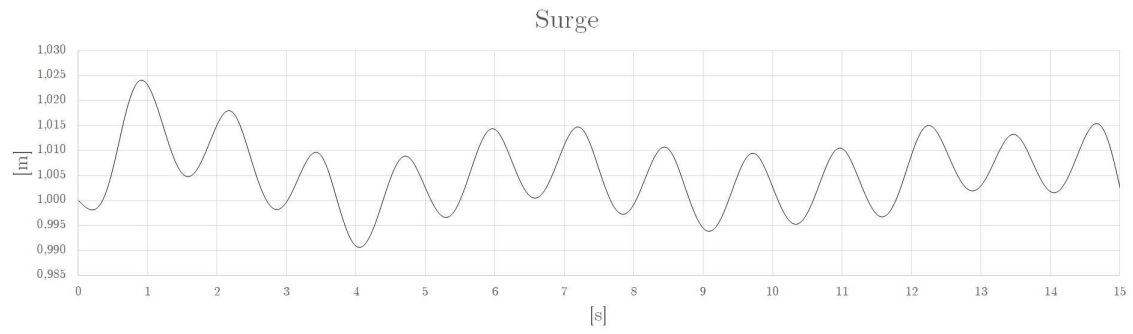


Figura 6.7: Caso A

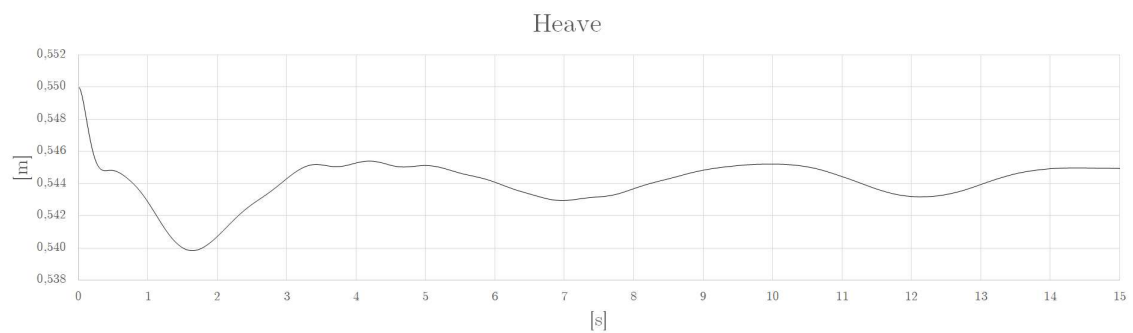
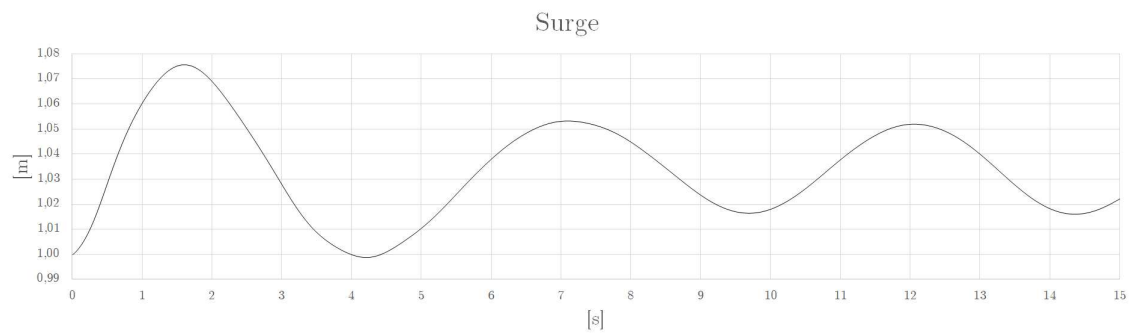


Figura 6.8: Caso A1

## 6. Modello preliminare 2D

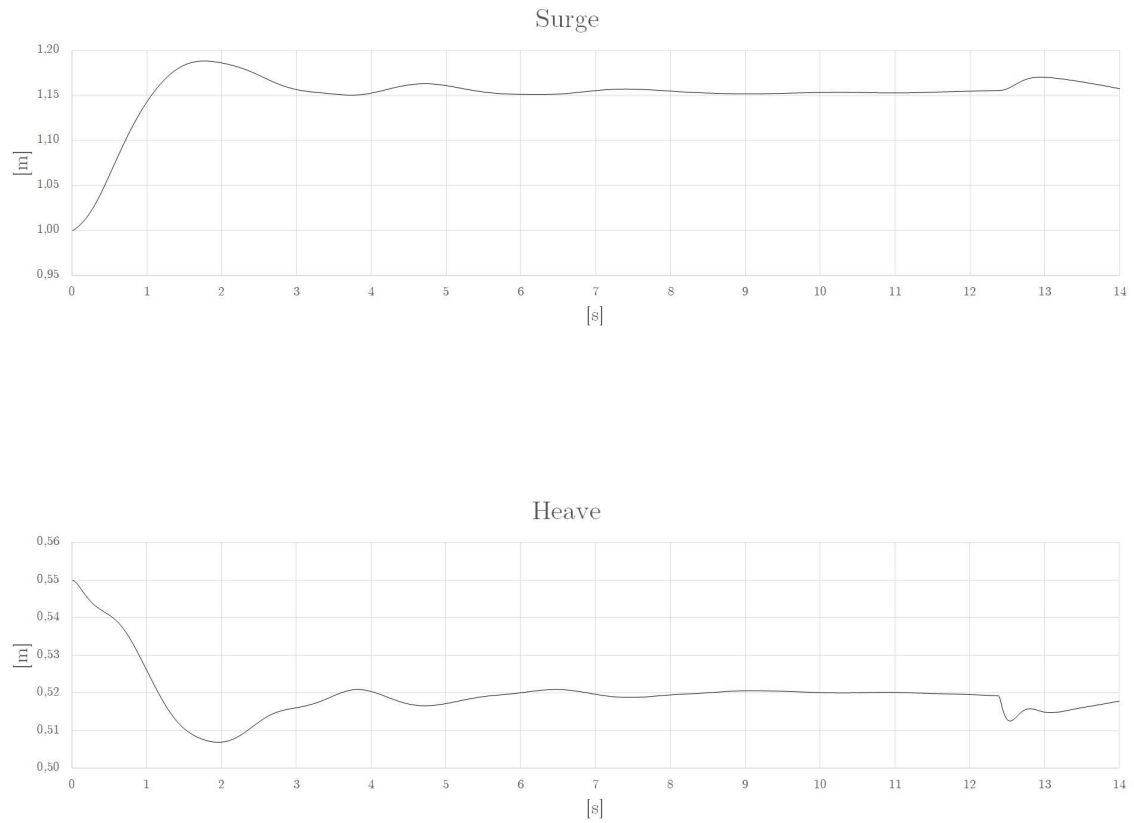


Figura 6.9: Caso A2

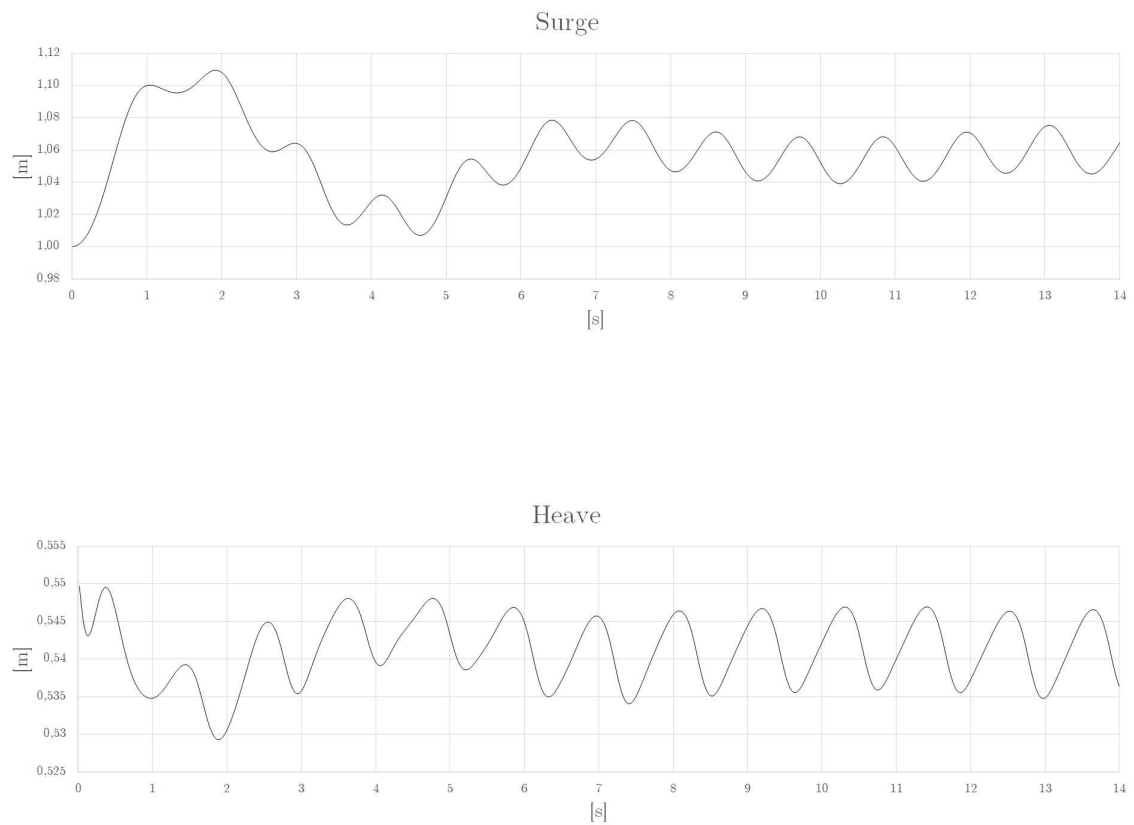


Figura 6.10: Caso A3

## 6. Modello preliminare 2D

---

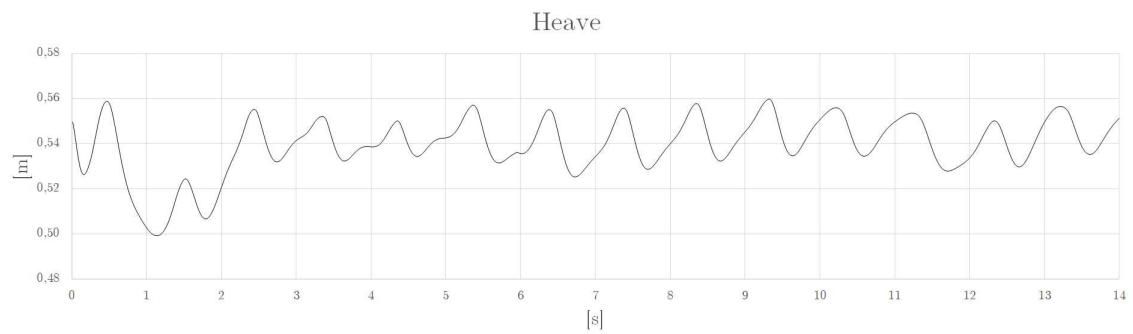
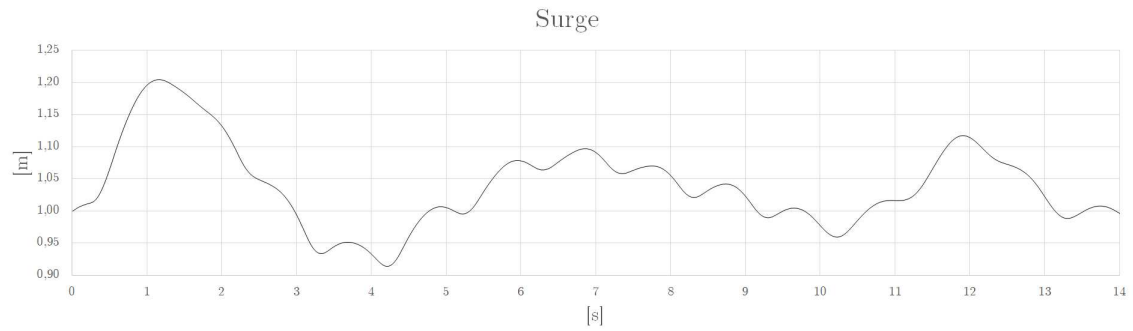


Figura 6.11: Caso A4

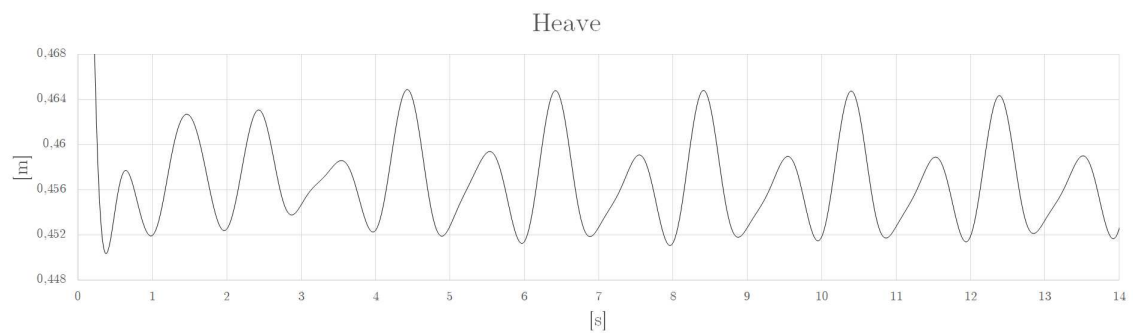
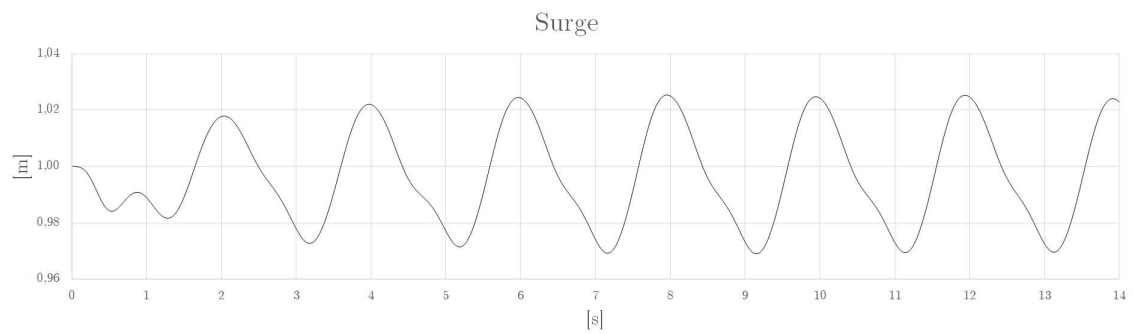


Figura 6.12: Caso B1

## 6. Modello preliminare 2D

---

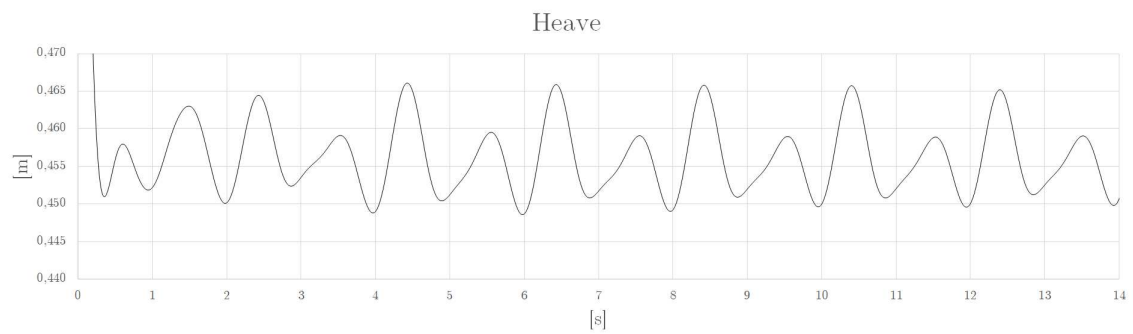
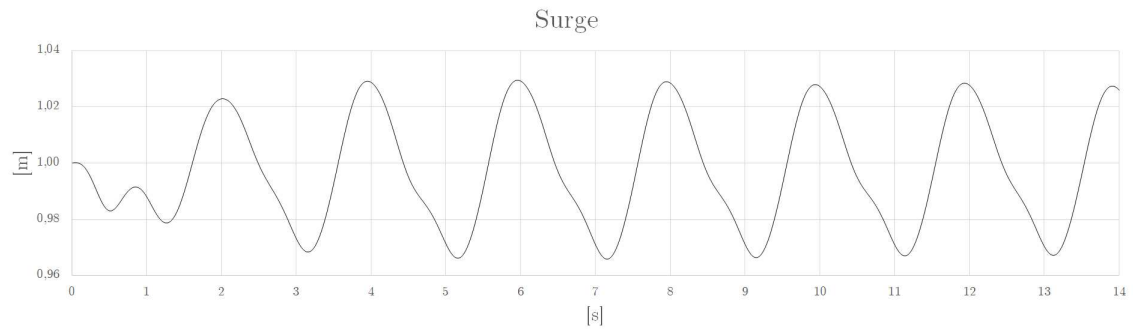


Figura 6.13: Caso B2

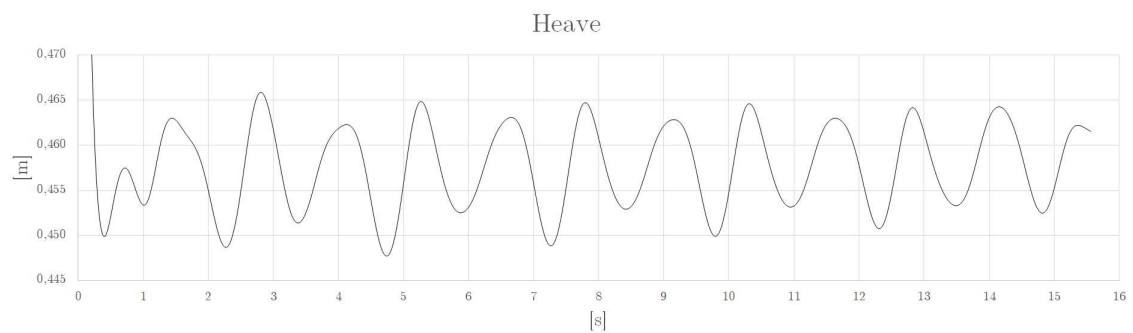
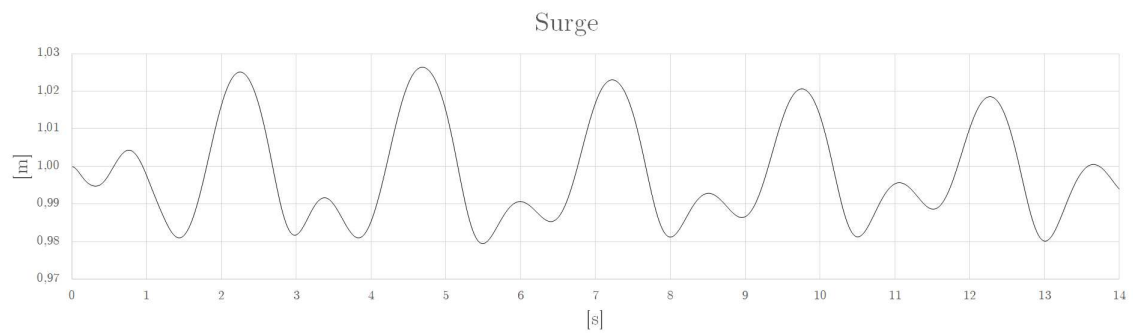


Figura 6.14: Caso B3



## 6. Modello preliminare 2D

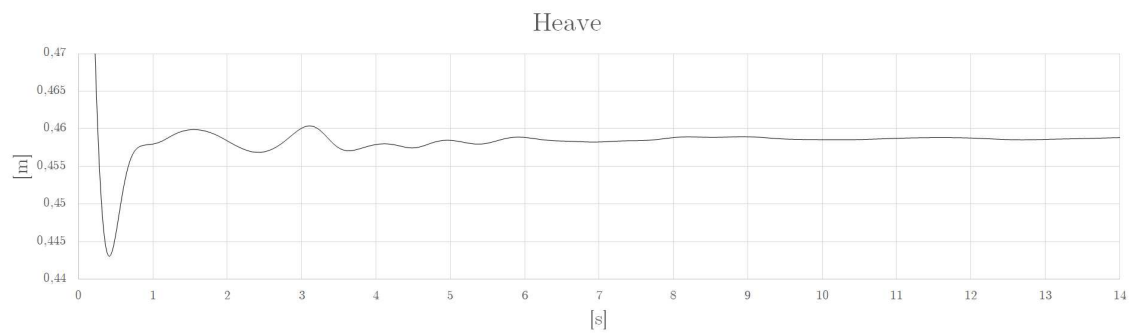
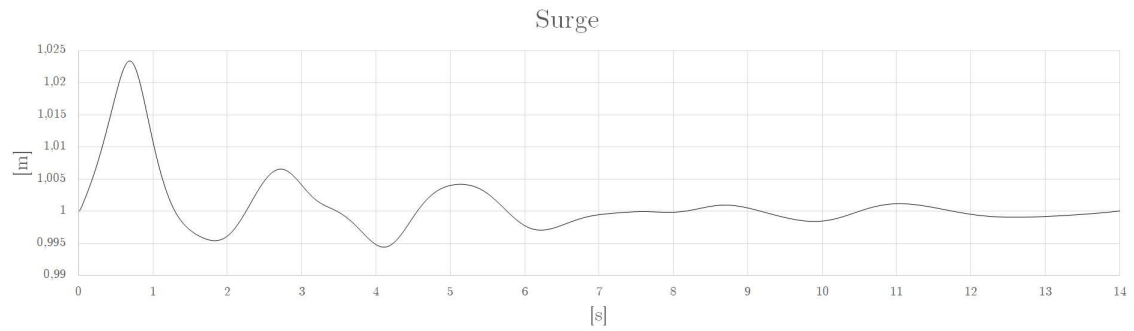


Figura 6.15: Caso B4

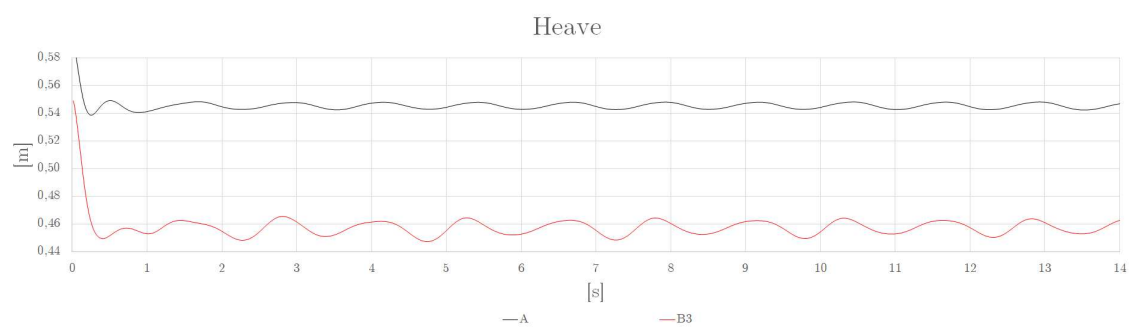
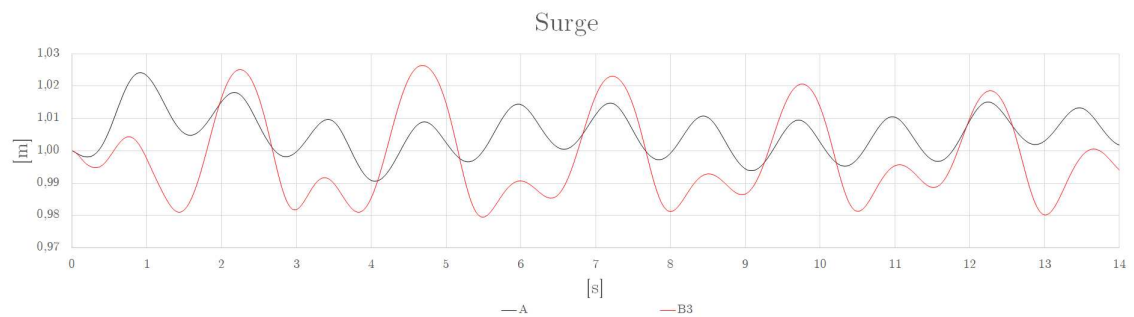


Figura 6.16: Casi A-B3

## 6. Modello preliminare 2D

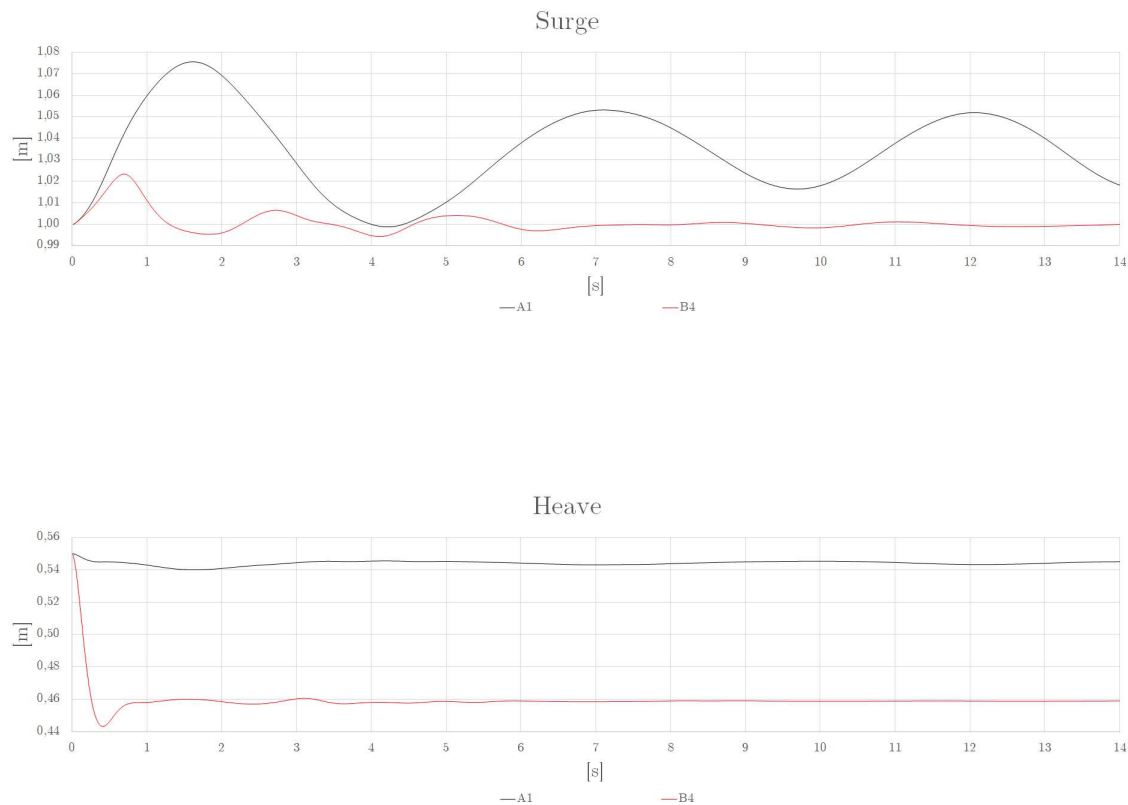


Figura 6.17: Casi A1-B4

Si nota dai risultati come il periodo dell'onda ha grande effetto sul comportamento dinamico della cassa in tutte e due le configurazioni degli ancoraggi.

È possibile vedere, inoltre, come le molle inclinate riducono gli spostamenti rispetto alla configurazione con molle verticali, in particolar modo quelli lungo la direzione di propagazione dell'onda (*Surge*).

Si deduce, quindi, che con gli ormeggi inclinati il sistema, in generale, è più stabile.

### 6.4 Comparazione con dati analitici

Si esegue ora uno studio analitico statico di alcuni casi simulati (uno per ogni tipo di corrente), semplificando gli effetti turbolenti e la rotazione della cassa (che nel caso in esame è minima); ci si limiterà a calcolare il

## 6. Modello preliminare 2D

punto di galleggiamento che scaturisce dall'equilibrio statico delle forze agenti sulla cassa (forza peso, spinta di Archimede, resistenza idrodinamica e forze delle molle), e da quello, la lunghezza finale analitica delle molle.

Quest'ultima verrà infine comparata con la lunghezza finale delle molle fuoriuscente dalle simulazioni numeriche. Per l'equilibrio delle forze orizzontali è necessario stimare la forza di resistenza prodotta dalle correnti agenti sulla cassa; si utilizza quindi, semplificando, la 2.21, dove al  $C_d$  viene assegnato un valore supposto pari a 1,5.

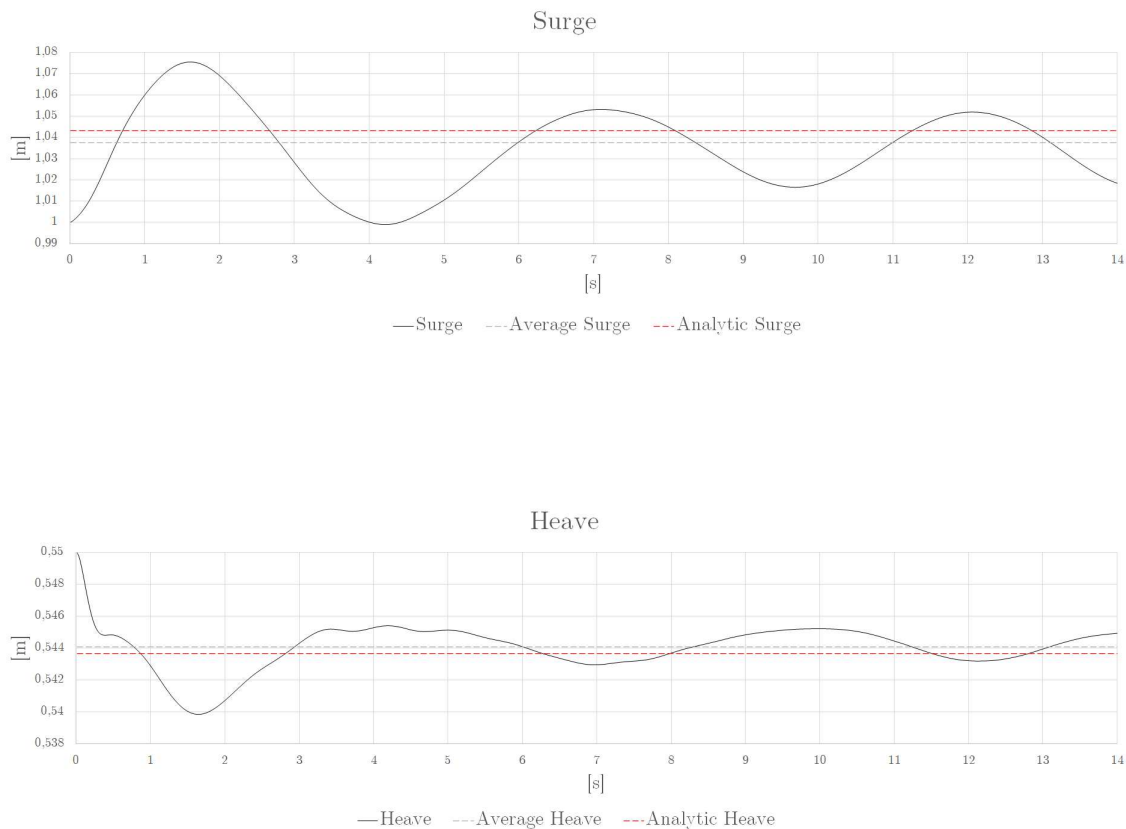


Figura 6.18: Comparazione caso A1 con dati analitici

## 6. Modello preliminare 2D

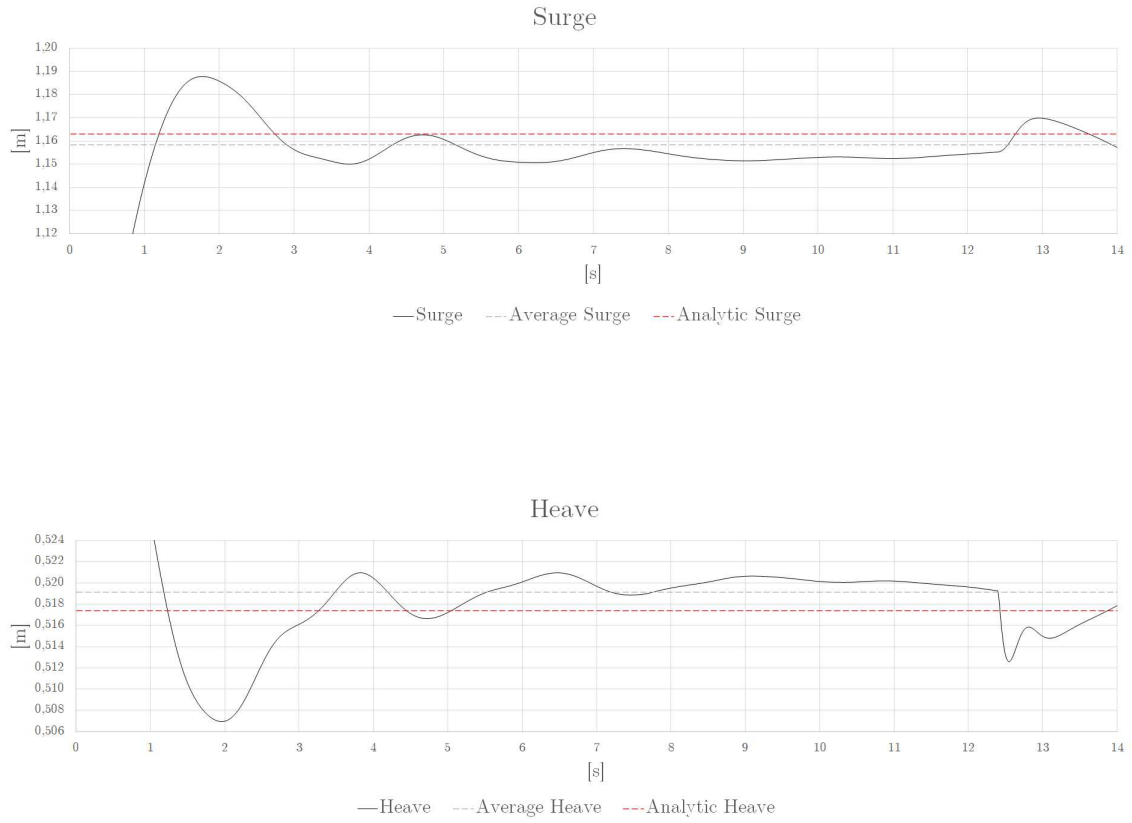


Figura 6.19: Comparazione caso A2 con dati analitici

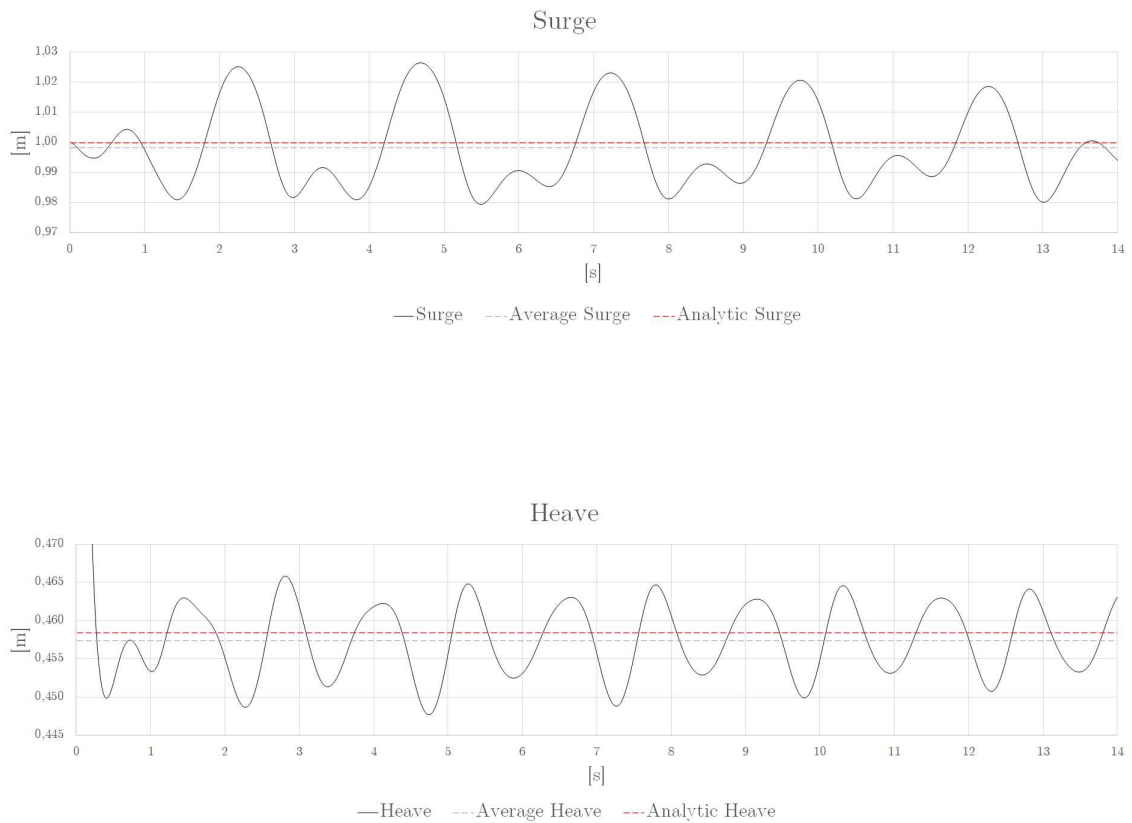


Figura 6.20: Comparazione caso B3 con dati analitici

## 6. Modello preliminare 2D

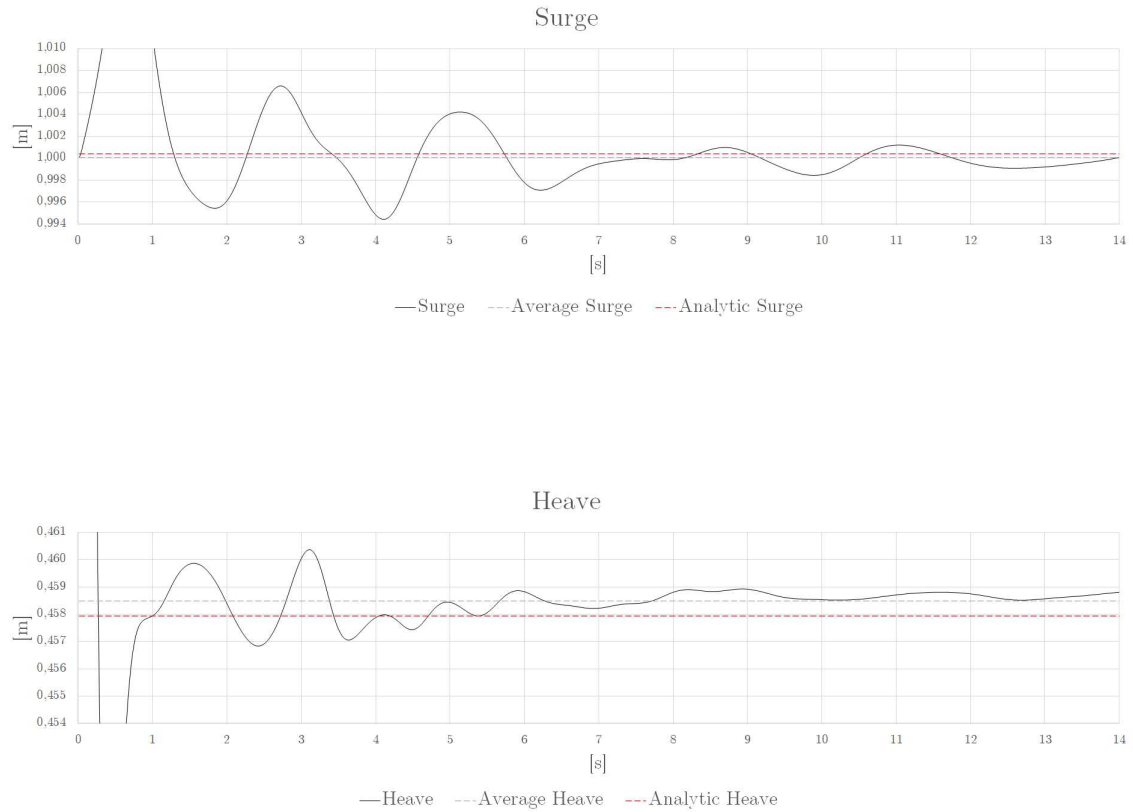


Figura 6.21: Comparazione caso B4 con dati analitici

Caso	Lunghezza	Lunghezza	Lunghezza	Errore
	iniziale	finale	finale	
	molla	analitica	numerica	
	[m]	[m]	[m]	[%]
<b>A1</b>	0.47	0.477103	0.477169	0.01
<b>A2</b>	0.47	0.477525	0.476498	0.22
<b>B3</b>	0.47	0.478685	0.477037	0.34
<b>B4</b>	0.47	0.478696	0.478976	0.06

Tabella 6.2: Comparazioni analitiche

I risultati analitici sono rappresentati dalla linea tratteggiata rossa, costante, poiché effetto di un equilibrio statico. Per comparare le simulazioni con tali dati si esegue una media nel tempo dei due parametri di galleggiamento, Heave e Surge (linea grigia tratteggiata).

## 6. Modello preliminare 2D

---

Essendo il moto ondoso regolare, infatti, il corpo dovrebbe oscillare attorno alla posizione di equilibrio che si avrebbe in assenza di onde (la corrente è invece presa in considerazione nel calcolo delle forze di drag). Sovrapponendo i risultati si nota come questi discostino, nel caso peggiore, di pochi millimetri; tale esito è perfettamente in linea con ciò che ci si aspettava: il modello analitico, infatti, non prende in considerazione gli effetti di bordo, della turbolenza, ed esegue una stima indicativa della forza di drag, ma dovrebbe prevedere in buona approssimazione il comportamento generale del corpo galleggiante, la posizione di equilibrio del suo centro di massa, e ciò è esattamente quello che accade.

Il modello 2D è quindi in grado di fornire risultati prevedibili partendo da delle condizioni iniziali customizzabili; il prossimo passo è quindi quello di passare a uno spazio tridimensionale, cambiando la geometria del dominio computazionale, del corpo galleggiante e modificando gli ancoraggi, così da ottenere un modello finale che simuli l'effettivo comportamento di una piattaforma eolica in condizioni effettive di mare aperto.

# 7

## Modello finale

Si presenta ora il modello 3D finale prodotto, un basamento per turbina eolica in scala 1:1, ancorato al fondale tramite taut-moorings, in condizioni di mare aperto. Al fine di poter installare nuovi pacchetti di librerie per la modellazione dei taut-moorings è stato necessario montare una versione più recente di OpenFOAM; tale passaggio modifica leggermente la scrittura del codice, soprattutto per la creazione delle onde e dei restraints, ma in generale i settaggi rimangono identici a quelli visti per il modello 2D. Il modello viene analizzato in varie condizioni di mare, con onde provenienti da diverse direzioni, così da evidenziare eventuali situazioni critiche. Vengono in seguito riportati i dettagli rilevanti del codice implementato.

### 7.1 Geometria

#### 7.1.1 Caratteristiche della piattaforma

La tipologia della piattaforma scelta per il modello è una *Barge-Based platform*, ed è caratterizzata da una base generalmente rettangolare con

## 7. Modello finale

---

una cavità passante nel centro. Non è tra le tipologie più diffuse, ma permette di ospitare turbine di dimensioni notevoli, e la cavità centrale del basamento consente una maggiore dissipazione delle oscillazioni, grazie alla porzione d'acqua 'intrappolata' dentro di essa.



Figura 7.1: Barge-Based platform

Per caricare la geometria di tale basamento su OpenFOAM, è necessario, come già visto in 4.2.2, creare un file in formato `.stl`, così che *snappyHexMesh* riesca a costruirci attorno una mesh esaedrica. Ciò è stato fatto con GstarCAD, partendo da una geometria data, estrapolandone un modello 3D. Una volta esportato il modello in formato `.stl`, basterà copiare il file nella cartella `constant/triSurface` ed eseguire *surfaceFeatureExtract* per importare la geometria nell'ambiente di lavoro di OpenFOAM.

Un passaggio fondamentale, quando si lavora con una geometria complessa, è quello di estrapolare le caratteristiche di quest'ultima, quali



## 7. Modello finale

---

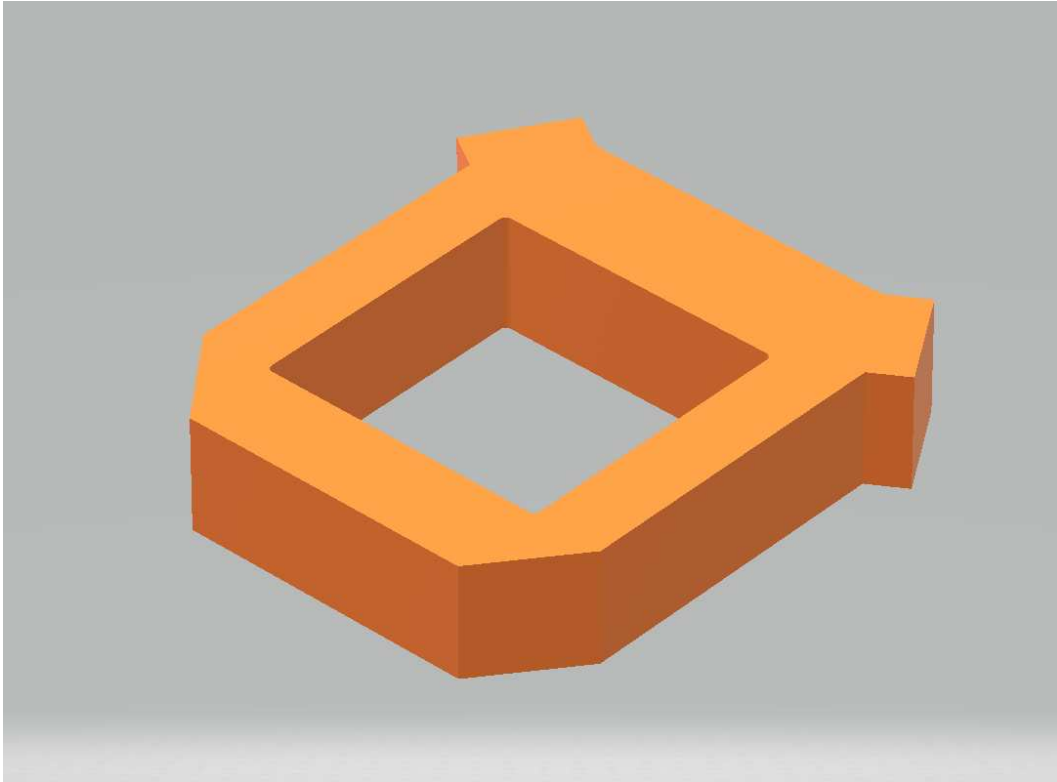


Figura 7.2: Modello 3D della Barge-Based platform con GstarCAD

centro di massa, tensore d'inerzia attorno al centro di massa, direzioni principali d'inerzia, parametri essenziali per la compilazione del file *dynamicMeshDict*. Per eseguire questo passaggio, un metodo veloce ed efficace è quello di utilizzare il tool *surfaceInertia*, specificando la densità dell'oggetto. Sovrascrivendo i risultati in un file ad hoc, avremo tutte le informazioni necessarie per il calcolo esatto della dinamica del corpo galleggiante a portata di mano.

```
Density: 350
Mass: 24535.3625716
Centre of mass: (4.68866761878 4.22402301571 0.959999978542)
Surface area: 169.847893599
Inertia tensor around centre of mass:
(152204.755434 0.00226115952273 -1.98951966013e-11
 0.00226115952273 204152.512128 1.98951966013e-11
```

## 7. Modello finale

---

```
-1.98951966013e-11 1.98951966013e-11 341595.158118)
eigenValues (principal moments): (152204.755434 204152.512128
341595.158118)
eigenVectors (principal axes):
(1 -4.35275682078e-08 1.05048604287e-16)
(4.35275682078e-08 1 -1.44752711882e-16)
(-1.05048597986e-16 1.44752716454e-16 1)
Transform tensor from reference state (orientation):
(1 4.35275682078e-08 -1.05048597986e-16 -4.35275682078e-08 1
1.44752716454e-16 1.05048604287e-16 -1.44752711882e-16 1)
Rotation tensor required to transform from the body reference
frame to the global reference frame, i.e.:
globalVector = orientation & bodyLocalVector

Writing scaled principal axes at centre of mass of "basamento.
stl" to "axes.obj"
```

Listing 7.1: Caratteristiche della geometria estratte tramite *surfaceInertia*

### 7.1.2 Mesh

La scelta dei parametri della mesh è di fondamentale importanza, poiché influisce sulla correttezza dei risultati: per simulare al meglio gli effetti turbolenti scaturiti dall'interazione solido-fluido è necessario rispettare specifiche condizioni, che rendano la mesh idonea. L'utilizzo di un modello  $K - \omega$  è consigliato in questo caso, poiché permette la risoluzione delle equazioni a parete senza raffinare la mesh in prossimità di queste ultime. Tuttavia la scelta di una discretizzazione spaziale molto raffinata ha un costo molto importante sulla velocità di avanzamento della simulazione e sulla pesantezza dei file in uscita. Nella costruzione di

## 7. Modello finale

---

un modello iniziale come questo, è più logico avanzare a piccoli step, partendo da una mesh molto grossolana, raffinandola man mano che si è sicuri di procedere nella giusta direzione. In questo modo si è in grado di effettuare tante simulazioni in breve tempo, con una potenza di calcolo limitata, con una memoria contenuta, e di concentrarsi su tutti gli altri aspetti del modello.

```
// Surface based refinement
// -----

// Specifies two levels for every surface . The first is the
// minimum level, every cell intersecting a surface gets refined
// up to the minimum level. The second level is the maximum
// level. Cell that 'see' multiple intersections where the
// intersections make an angle > resolveFeatureAngle get refined
// up to the maximum level.

refinementSurfaces
{
    basamento
    {
        // Surface-wise min and max refinement level
        level (4 4);
    }
}

resolveFeatureAngle    10;
```

Listing 7.2: Estratto di codice da *snappyHexMesh* per il raffinamento della mesh attorno all'oggetto

Nel caso in esame il dominio computazionale è un parallelepipedo delle

## 7. Modello finale

---

dimensioni di 190  $m$  lungo  $X$ , 190  $m$  lungo  $Y$  e 130  $m$  lungo  $Z$ , e ha una discretizzazione rispettivamente di 80, 80 e 50 elementi. Una suddivisione di questo tipo è evidentemente molto grossolana, ma in queste fasi, come già detto, ci si concentra sulla funzionalità del modello più che sulla sua accuratezza. In futuro, con maggiori potenze di calcolo e memoria si procederà al raffinamento della mesh.

Un metodo efficace per eseguire quest'ultima operazione è quello di eseguire un raffinamento locale, mantenendo il dominio di fondo più grossolano; una volta inserita la geometria, infatti, è possibile affinare la mesh solamente attorno all'oggetto, modificando dei parametri nel file *snappyHexMesh* (7.2).

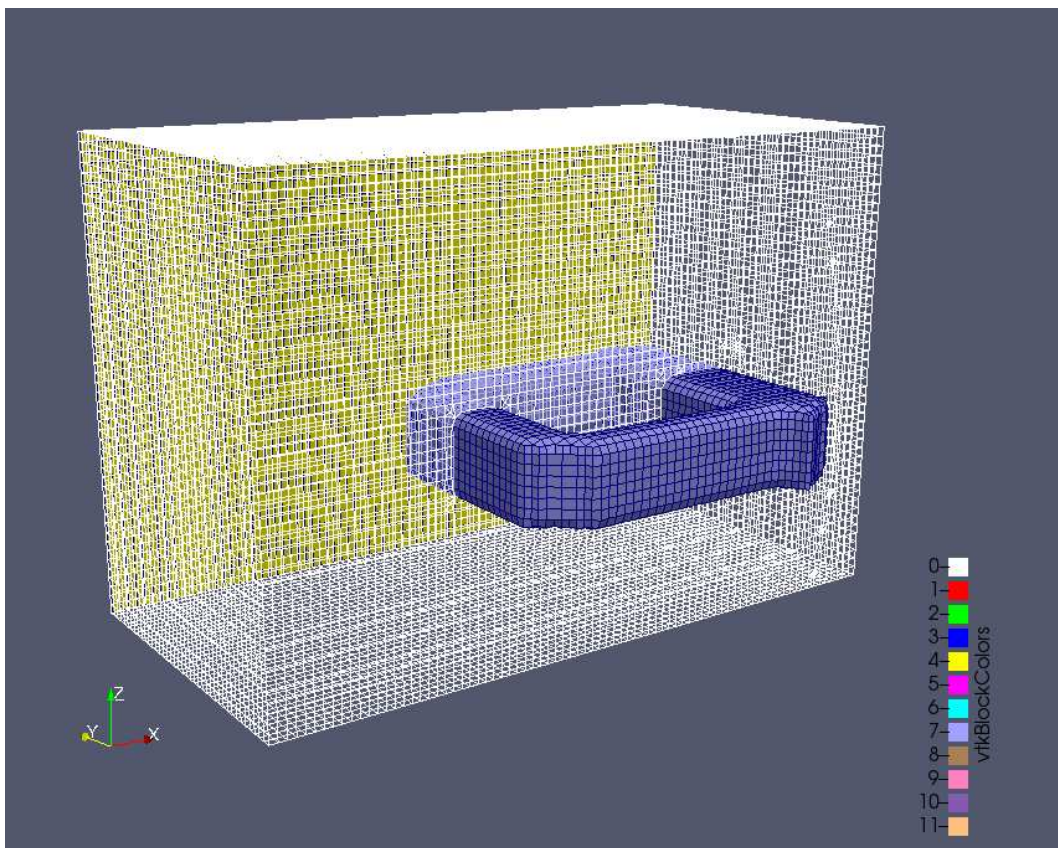


Figura 7.3: Rappresentazione della mesh del modello finale

## 7. Modello finale

### 7.1.3 Moody

Il nuovo modello 3D è ancorato al suolo tramite taut-moorings, non più con dei sistemi massa-molla-smorzatore. Per simulare con efficacia l'effetto di tali ancoraggi, è stato necessario implementare nel codice OpenFOAM una libreria esterna, specifica per ormeggi dinamici, *Moody*. Questa libreria, anche se indipendente, permette di essere accoppiata ad altri software di simulazione, tra i quali OpenFOAM; in particolare, una volta installata, essa andrà a generare una nuova classe di *restraints* personalizzabili, come catenarie o taut-moorings.

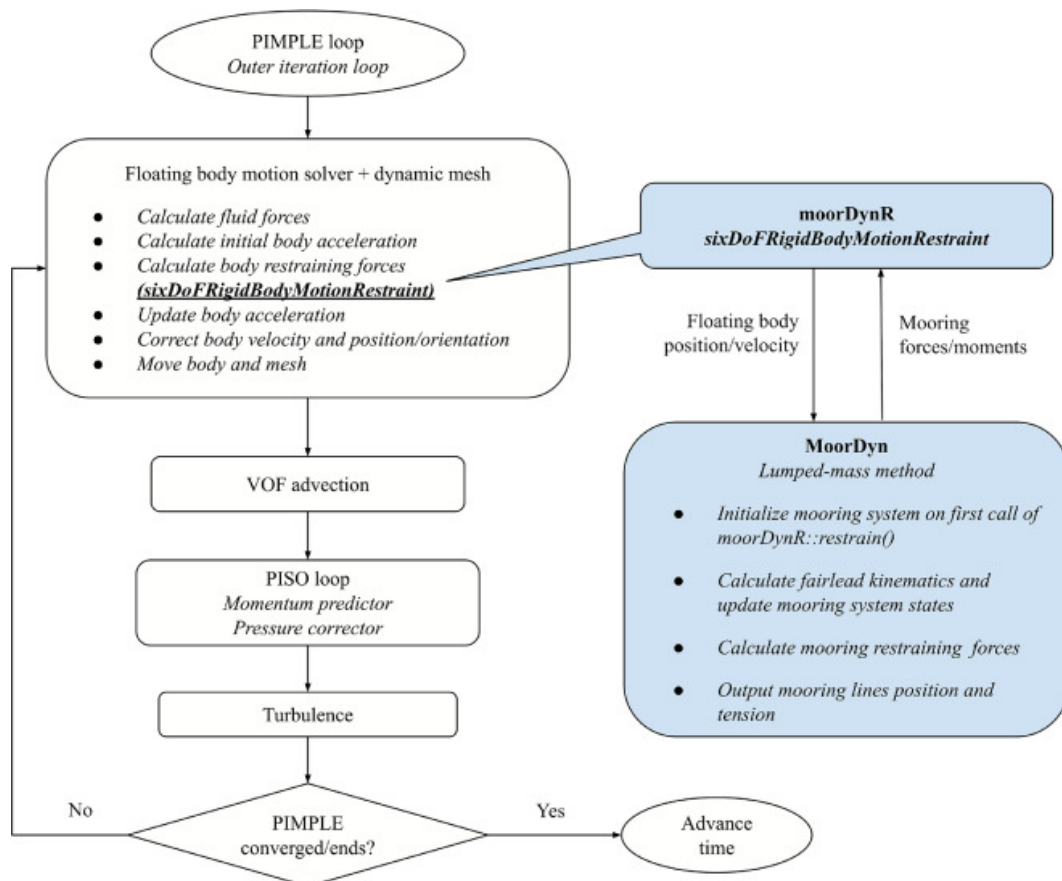


Figura 7.4: Ciclo di iterazione con l'utilizzo di Moody

L'utilizzo di questa libreria, in sostanza, si traduce nell'aggiunta della cartella *'mooring'* all'interno della directory principale; all'interno

## 7. Modello finale

---

di essa, nel file *OFmooring.m* (precompilato) saranno descritte tutte le caratteristiche degli ormeggi utilizzati: tipologia di ancoraggio, caratteristiche fisiche, punti di attacco, tipo di discretizzazione, condizioni al contorno degli attacchi (7.4). All'interno di questo file saranno indicati, in particolare, i punti di attacco delle linee di ormeggio al fondale, mentre i punti fissati sul corpo galleggiante vengono specificati come punti esterni al Moody; questi ultimi verranno descritti direttamente nel file *dynamicMeshDict* (7.3).

```
restraints
{
    moodyRestraint
    {
        type    moodyR;
        body    basamento;
        attachmentPoints
        (
            (42.24023 0.1 1)
            (19.24023 84.25 1)
            (65.24023 84.25 1)
        );
        mooringSystemFileName    "mooring/OFmooring.m";
    }
}
```

Listing 7.3: Estratto di codice da *dynamicMeshDict* - accoppiamento con Moody

In questa fase del progetto, le caratteristiche fisiche degli ormeggi sono descritte arbitrariamente; una volta appurata la funzionalità del modello si procederà alla definizione specifica dell'ancoraggio in ogni suo aspetto.

## 7. Modello finale

---

```
// Cable type:
cableType1.diameter = 0.025;
cableType1.rho = 5000;
cableType1.gamma0 = 0.5;
cableType1.materialModel.type = 'bilinear';
cableType1.materialModel.EA = 8000000; // Mean axial stiffness
cableType1.materialModel.xi = 0.5; // Damping coefficient

// Optional input of hydrodynamic coefficients
cableType1.CDn = 1.6; // Drag force coefficient
cableType1.CDt = 0.05;
cableType1.CMn = 1.0; // Added mass coefficient
cableType1.CMt = 0;

// Boundary conditions
bc1.vertexNumber = 1;
bc1.type = 'dirichlet';
bc1.mode = 'fixed';
bc2 = bc1;
bc2.vertexNumber = 2;
bc3.vertexNumber = 1;
bc3.type = 'dirichlet';
bc3.mode = 'external';
bc4 = bc3;
bc4.vertexNumber = 4;

// API connectivity
API.bcNames = {'bc3', 'bc4'};
```

Listing 7.4: Estratto di codice da *OFmooring.m* - caratteristiche fisiche degli ancoraggi

### 7.2 Condizioni iniziali

La Barge-based platform è mantenuta in posizione da tre linee di ancoraggio, taut-moorings posizionate come in figura 7.5: un ancoraggio nella parte frontale e due nella parte posteriore, angolati di  $60^\circ$  rispetto al primo, con una lunghezza iniziale di 58.6 m e un pretensionamento dello 0.07 %.

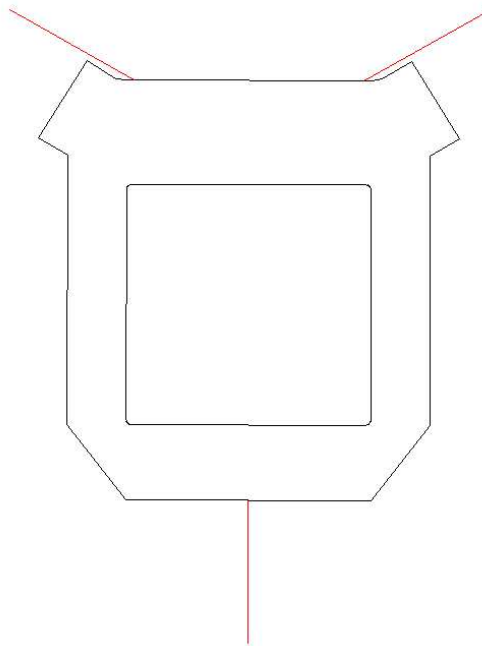


Figura 7.5: Posizionamento taut-moorings

Il fondale è posizionato a una profondità di 50 m, e le condizioni di mare simulate sono due:

~ una configurazione con un periodo di ritorno di un anno,  $H = 8m, T_s = 9s$ ;

~ una configurazione con un periodo di ritorno di cento anni,  $H = 13m, T_s = 11s$ .

Per ogni condizione di mare sono state simulate due angoli di incidenza,  $0^\circ$  e  $90^\circ$  rispetto alla direzione dell'ancoraggio frontale.



## 7. Modello finale

### 7.3 Risultati

Vengono ora mostrati tutti i grafici derivanti dalle simulazioni effettuate, comparando poi i risultati delle varie configurazioni. La convenzione sui segni è riportata in figura 6.5; si ricorda che *Pitch* è l'unica rotazione possibile, ed è riportata nei risultati come *Rotation*

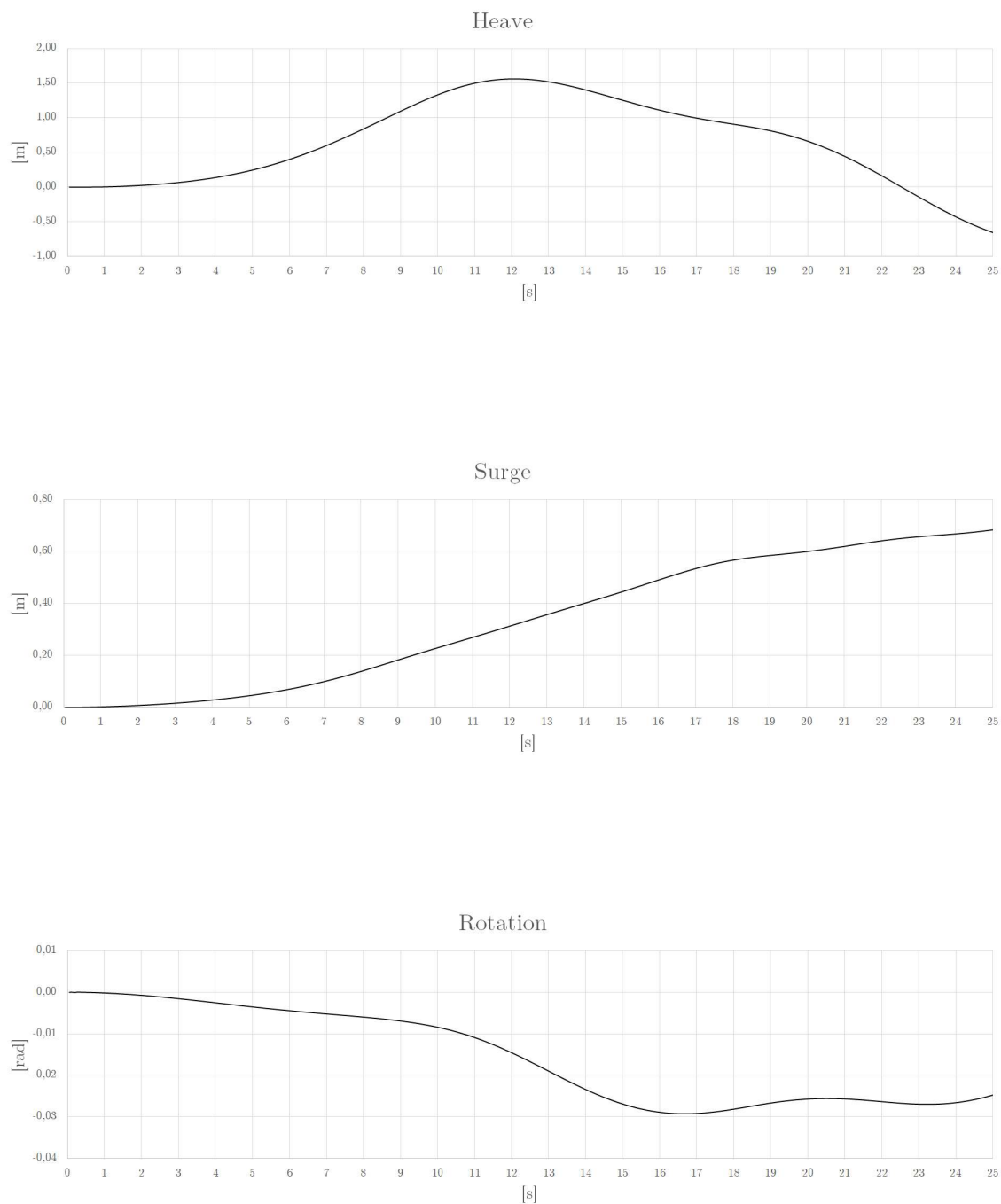


Figura 7.6: Onda 1 anno, direzione 0 gradi

## 7. Modello finale

---

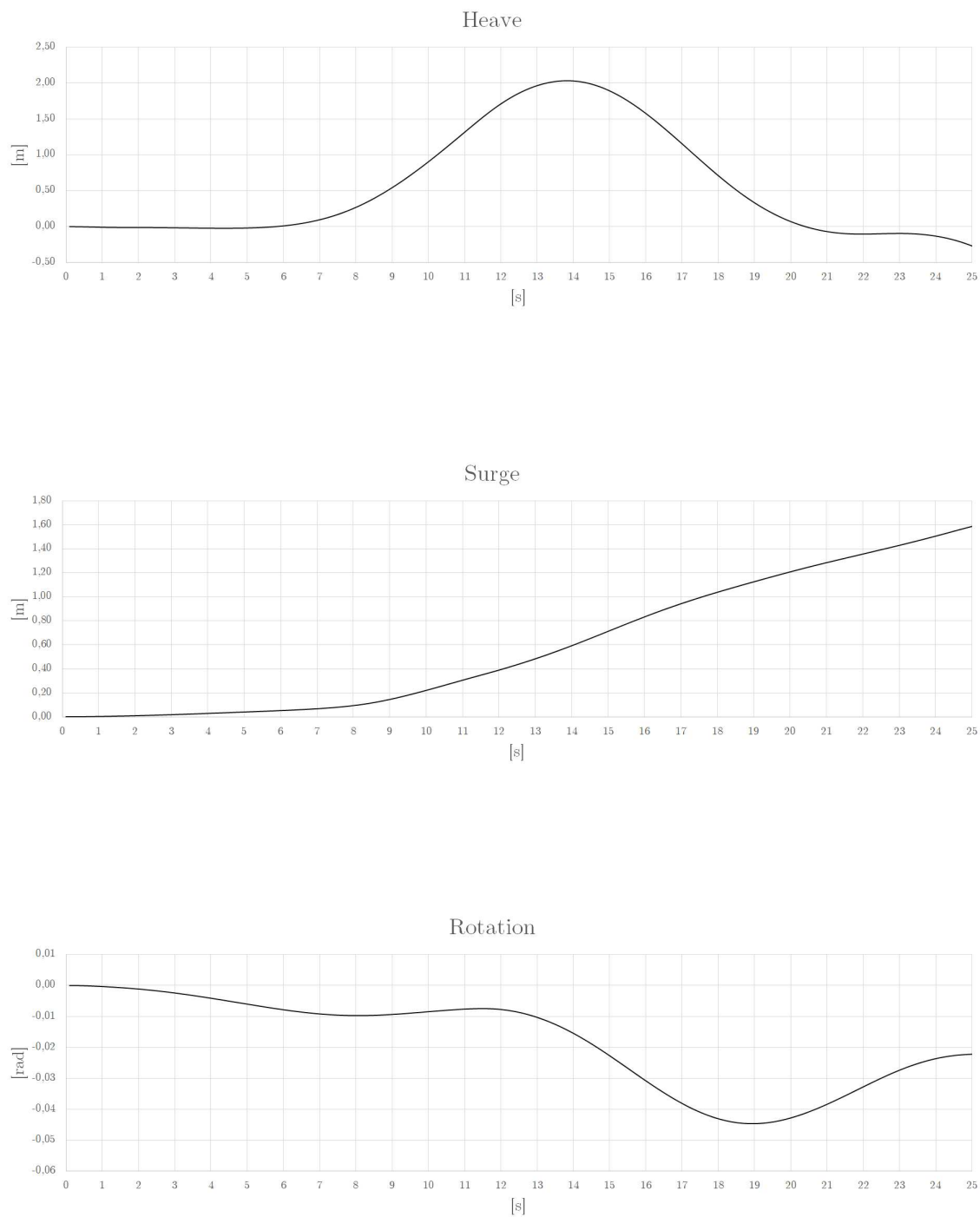


Figura 7.7: Onda 100 anni, direzione 0 gradi

## 7. Modello finale

---

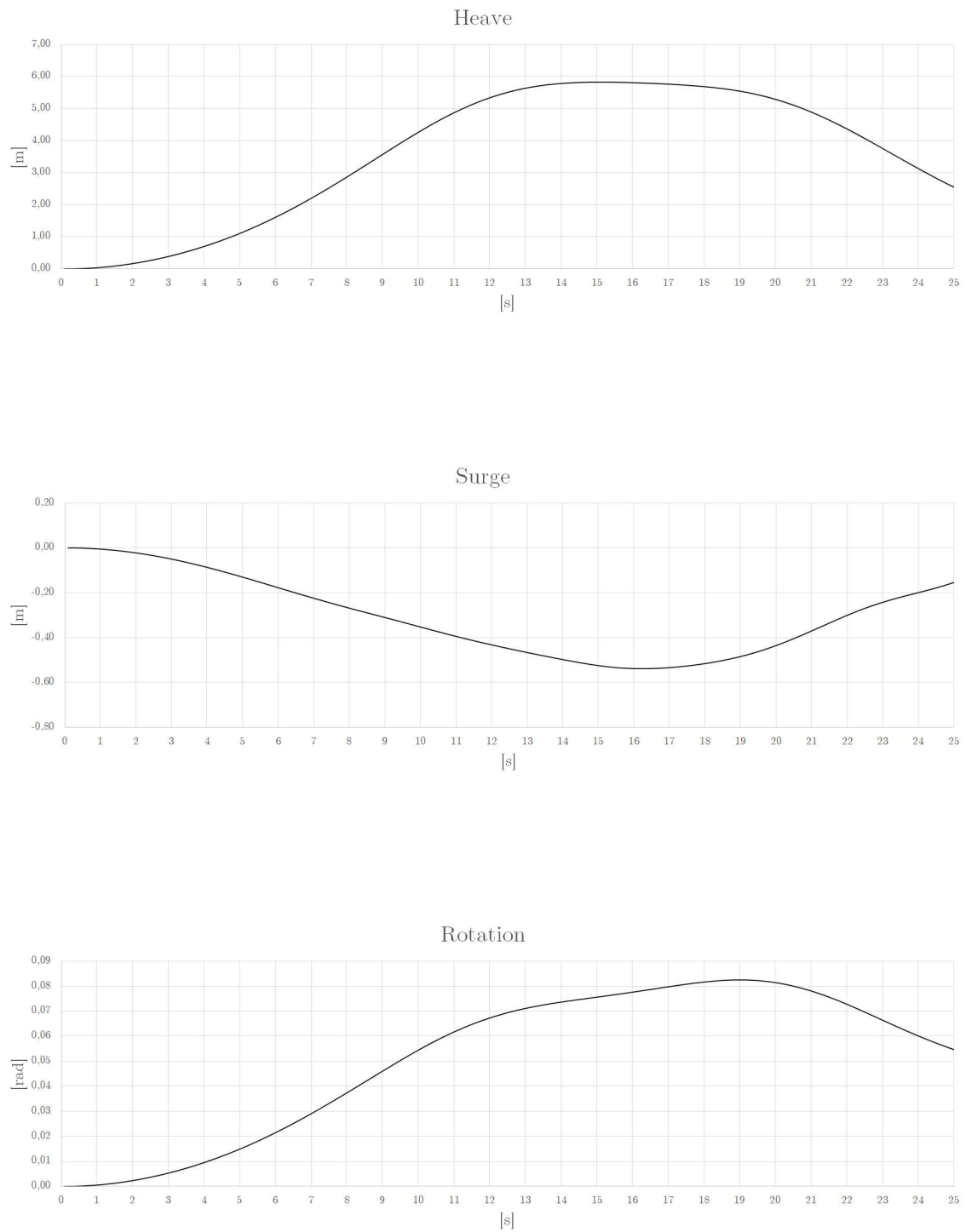


Figura 7.8: Onda 1 anno, direzione 90 gradi

## 7. Modello finale

---

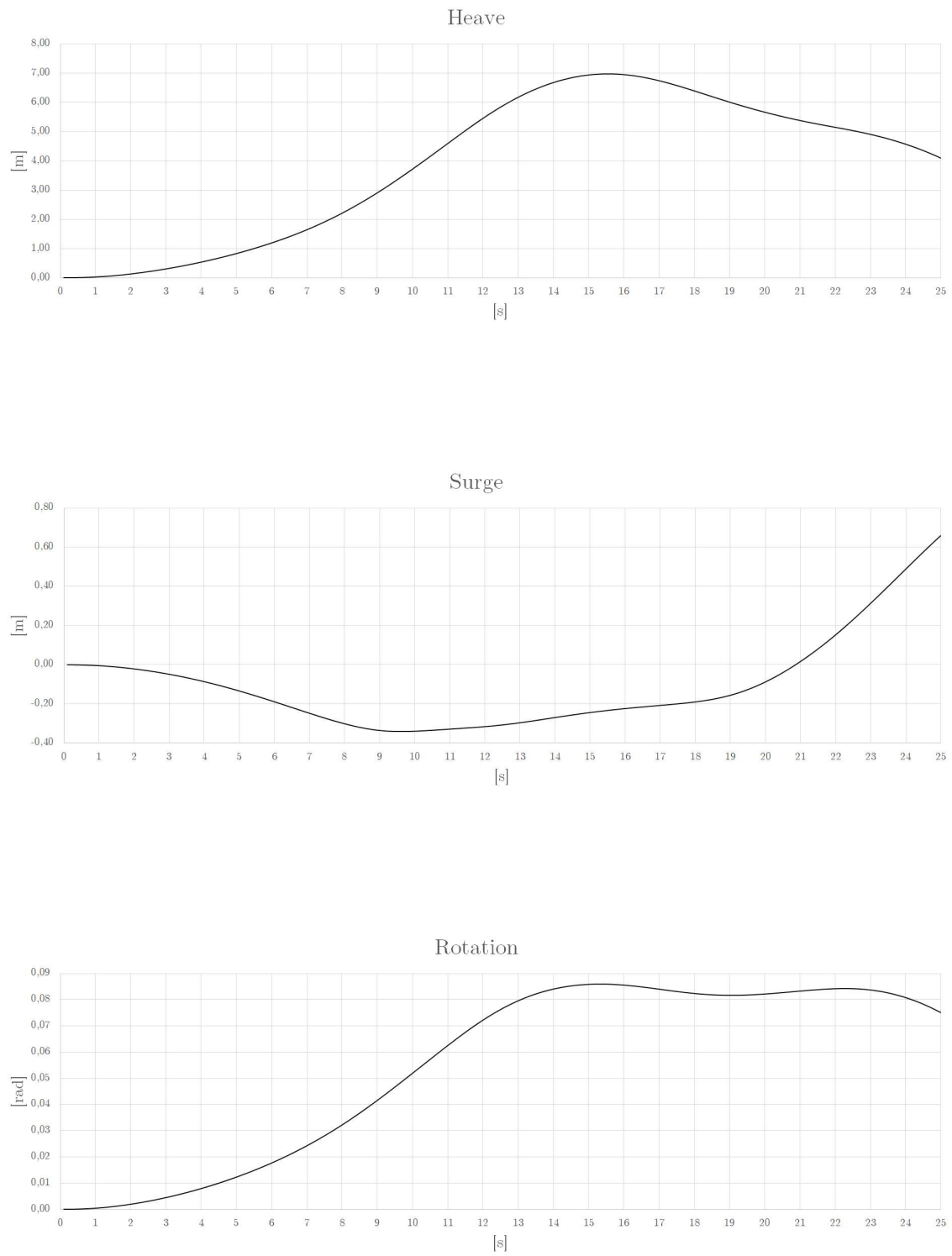


Figura 7.9: Onda 100 anni, direzione 90 gradi

## 7. Modello finale

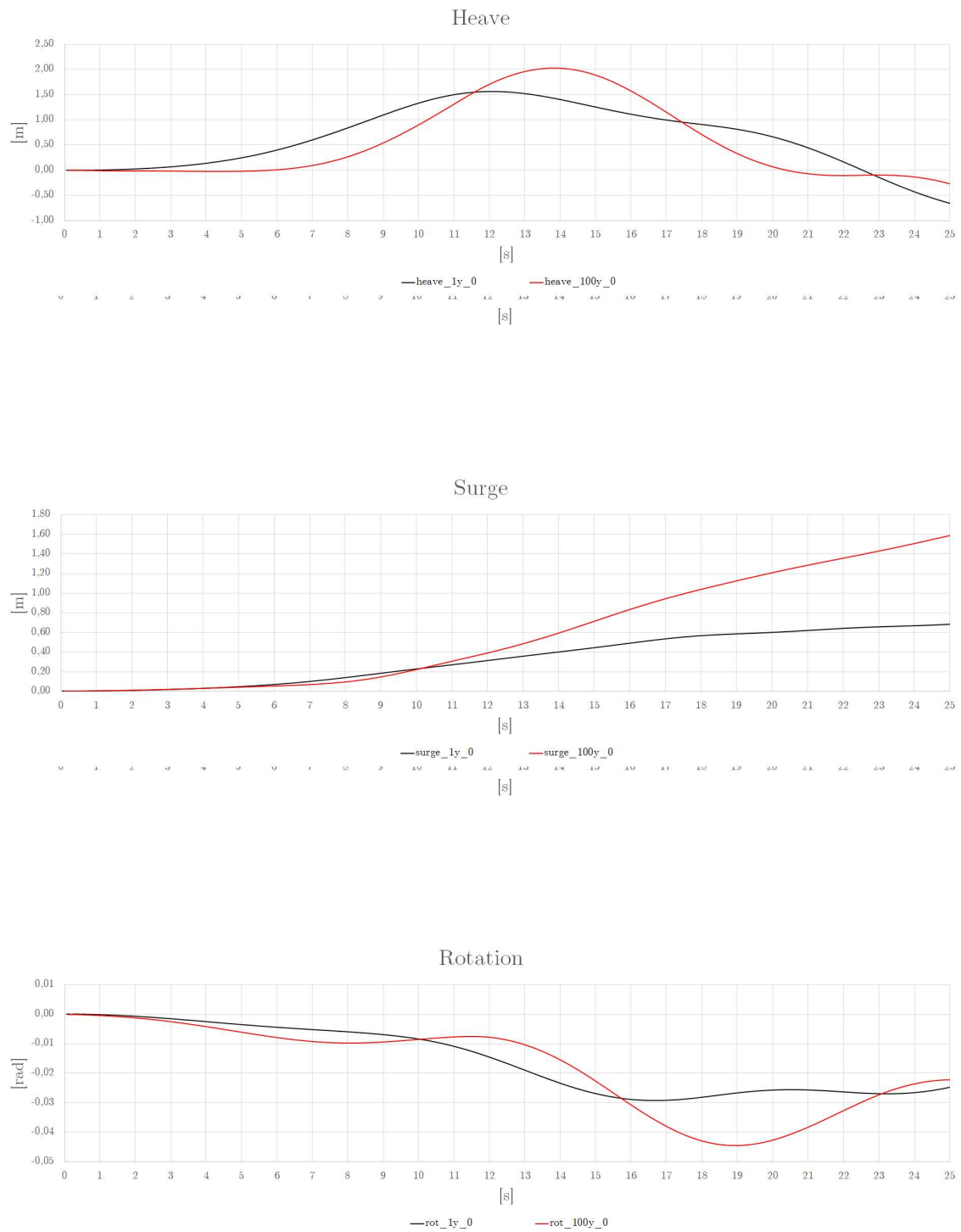


Figura 7.10: Direzione 0 gradi

## 7. Modello finale

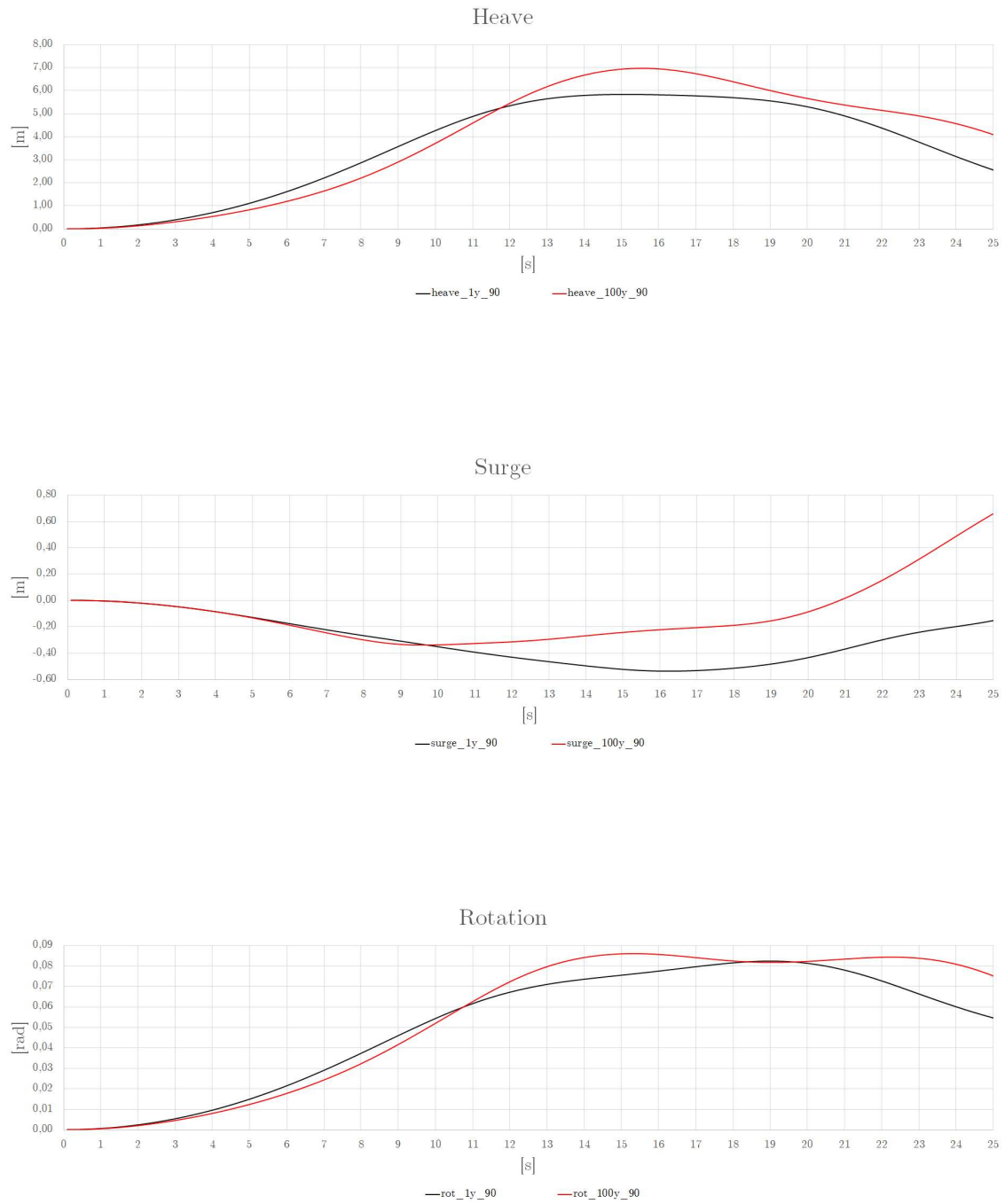


Figura 7.11: Direzione 90 gradi

## 7. Modello finale

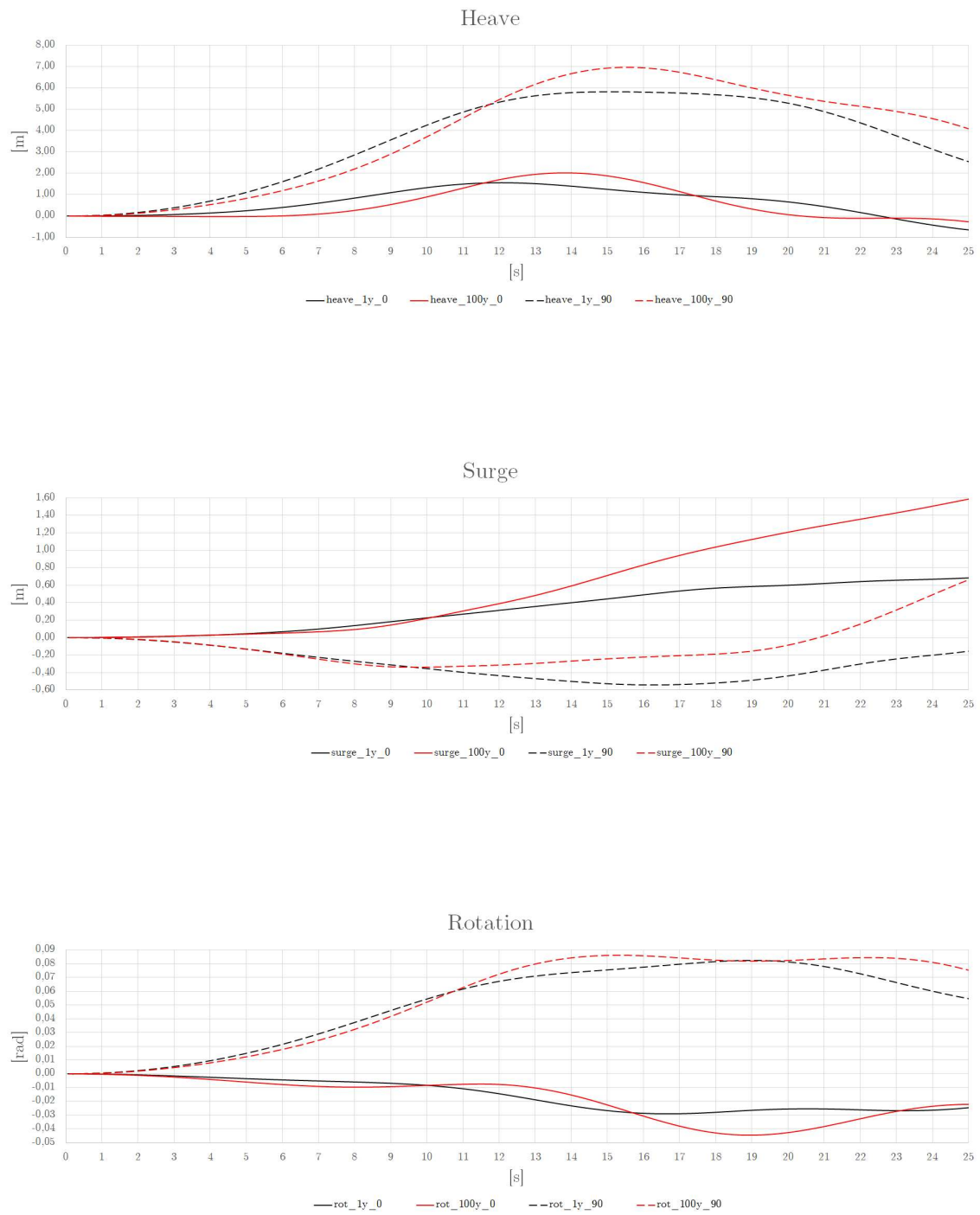


Figura 7.12: Comparazione dei 4 casi analizzati

## 7. Modello finale

---

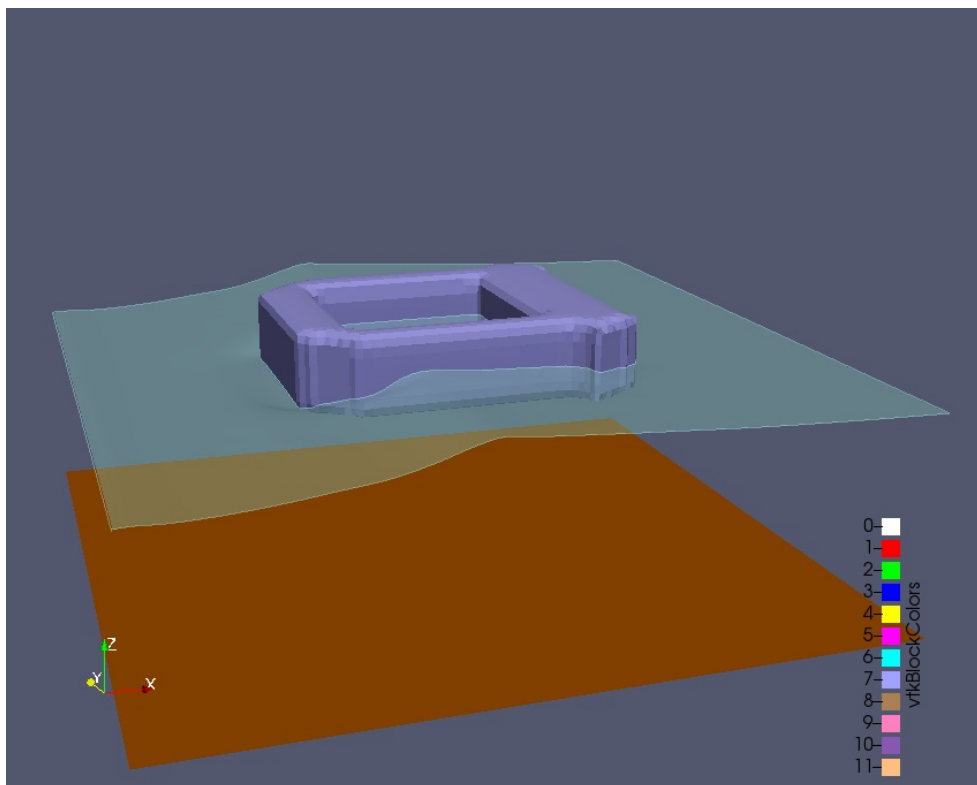


Figura 7.13: Snapshot 1 - Onda 100 anni, direzione 0 gradi

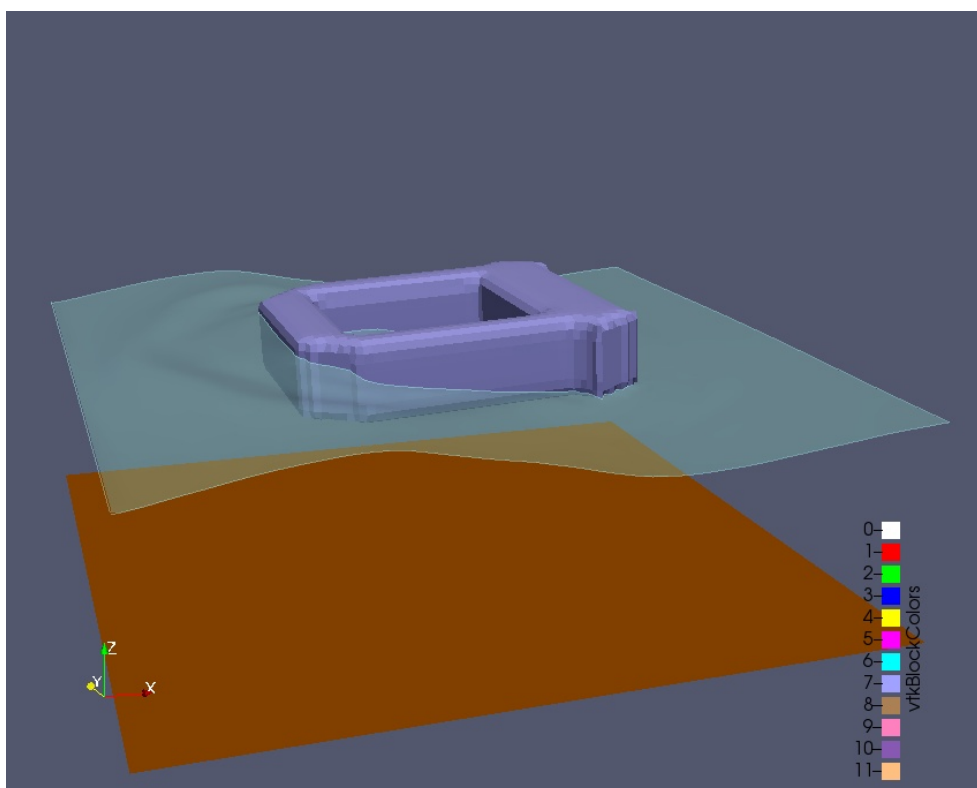


Figura 7.14: Snapshot 2 - Onda 100 anni, direzione 0 gradi



## 7. Modello finale

---

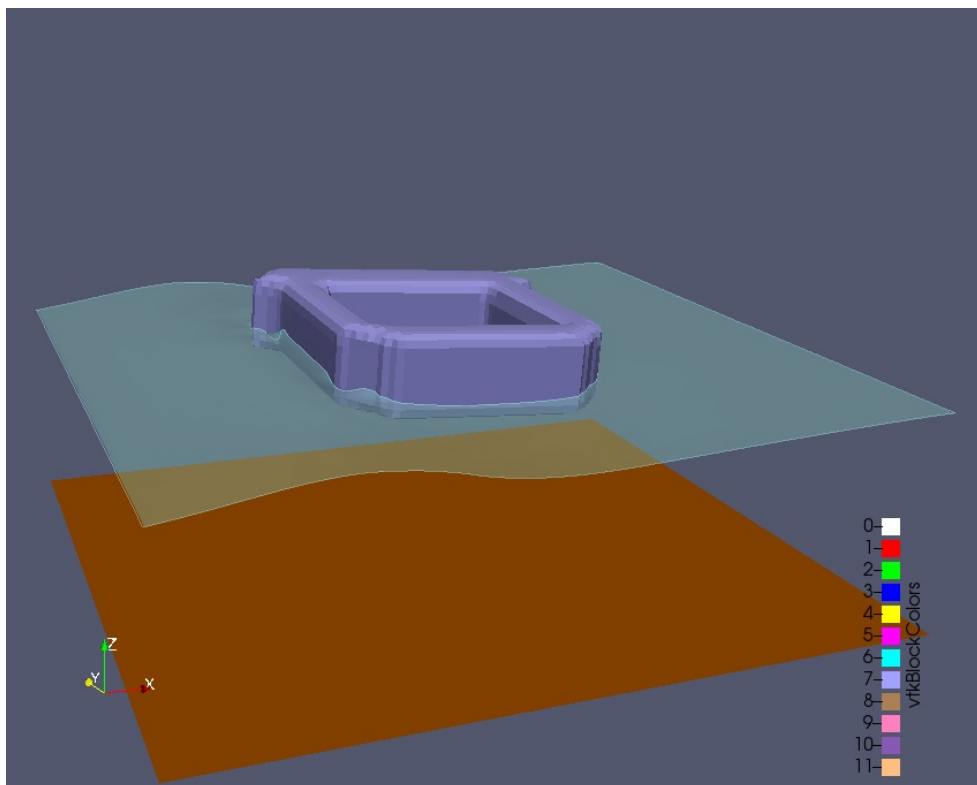


Figura 7.15: Snapshot 3 - Onda 1 anno, direzione 90 gradi

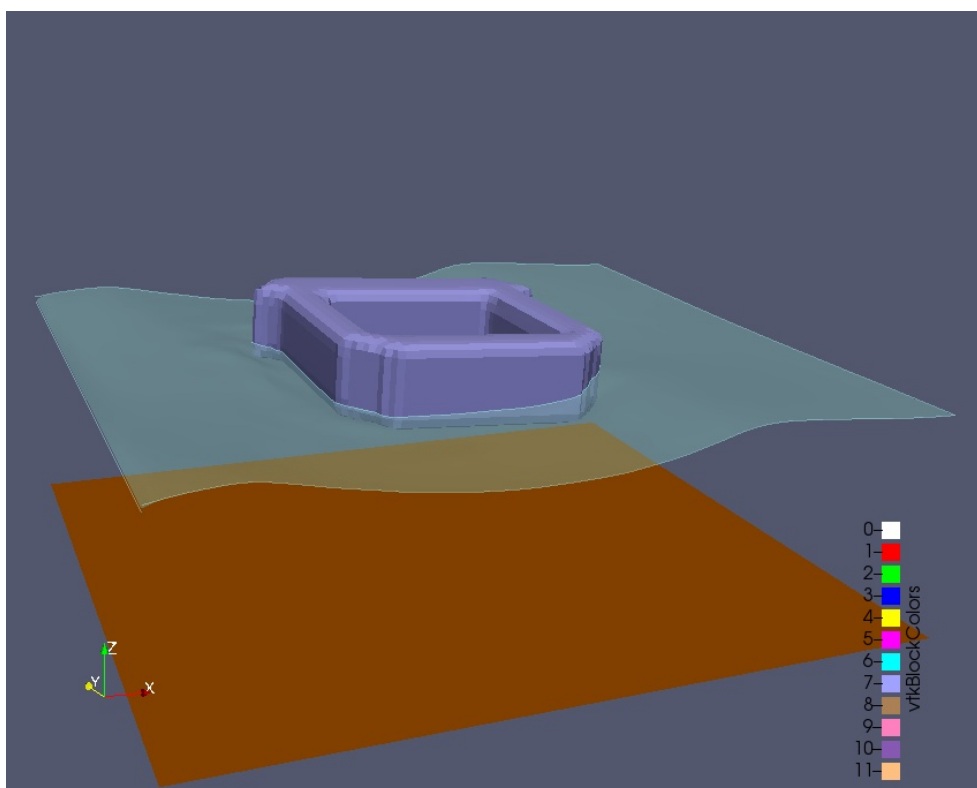


Figura 7.16: Snapshot 4 - Onda 1 anno, direzione 90 gradi

## 7. Modello finale

---

Dai risultati è possibile osservare che:

- ~ la differenza più grande si osserva comparando le due direzioni con le quali le onde impattano sulla struttura; sebbene, con onde a  $90^\circ$  la Surge rimane contenuta, l'Heave ha un picco più di 3 volte maggiore rispetto ad un'onda a  $0^\circ$ , e la Rotation è circa due volte maggiorata. Grazie a ciò è possibile dedurre che, con tale configurazione degli ancoraggi, la piattaforma assorbe meglio onde frontali rispetto a quelle laterali; analizzando quindi le onde caratteristiche della zona in cui si vuole posizionare la turbina, si potrà estrapolare una direzione preferenziale di arrivo delle onde, e orientare il basamento parallelamente ad essa;
- ~ osservando i due tipi di onda provenienti dalla stessa direzione, il parametro più sensibile è il Surge, che si modifica in ampiezza con un rapporto di circa 1:1 rispetto all'incremento in ampiezza dell'onda; anche l'Heave è molto sensibile all'ampiezza dell'onda, aumentando di circa il 25% tra la prima e la seconda onda; per la Rotation si osserva una cosa diversa: essa è più sensibile quando l'onda interagisce in modo frontale con la struttura, mentre lateralmente raggiunge all'incirca gli stessi valori massimi.

Le osservazioni che si possono effettuare sono comunque limitate a pochi casi simulati, per le ragioni di cui si è già parlato; risulta quindi difficile in questo momento delineare un trend di risposta della struttura ai carichi marittimi. Analizzando più casi, più direzioni e più frequenze d'onda per ogni direzione si potrà effettivamente comprendere appieno il modo in cui la struttura risponde a tali forzanti esterne; per ora è importante sottolineare la funzionalità del modello, il quale può infatti simulare questo tipo di situazioni, riportando dei risultati comprensibili.

## 7. Modello finale

---

Un'ultima menzione è necessaria per esplicitare tutte le potenzialità del modello, al di là che queste siano state o meno analizzate nell'ambito di questa tesi: grazie al visualizzatore di postprocessing *paraView* è possibile osservare il comportamento di parametri quali velocità, pressione e turbolenza, così da avere una visuale completa sull'interazione fluido-struttura. Vengono in seguito riportati alcuni snap dei campi di velocità, pressione, turbolenza sull'interfaccia multifase e struttura.

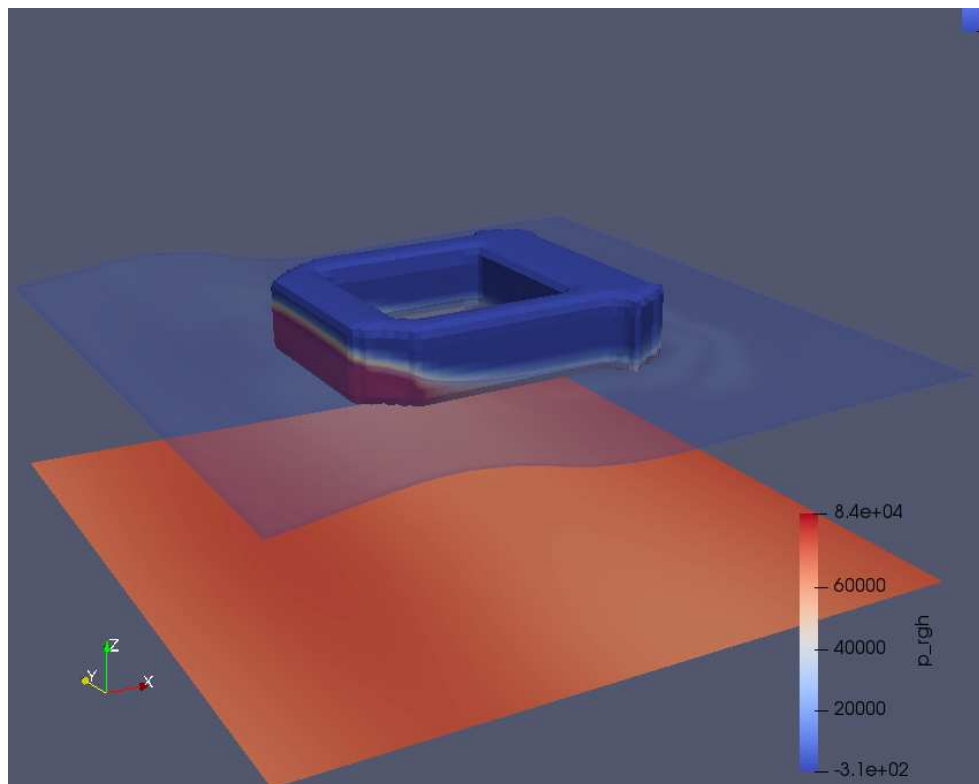


Figura 7.17: Postprocessing - pressione sulla struttura

## 7. Modello finale

---

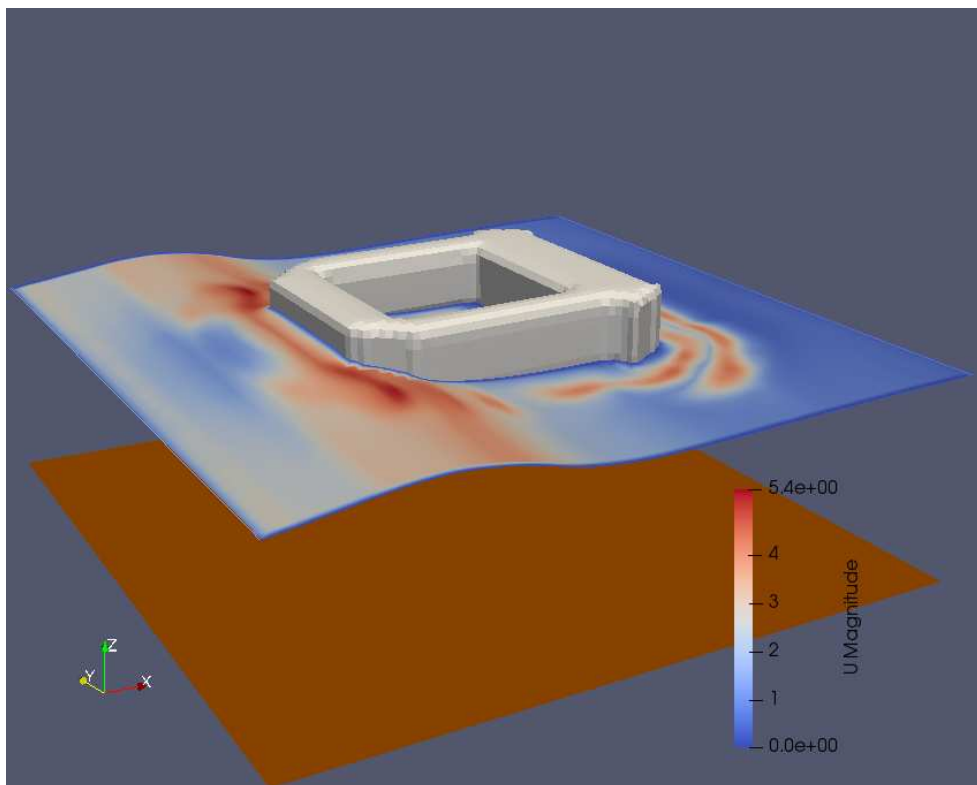


Figura 7.18: Postprocessing - velocità del fluido

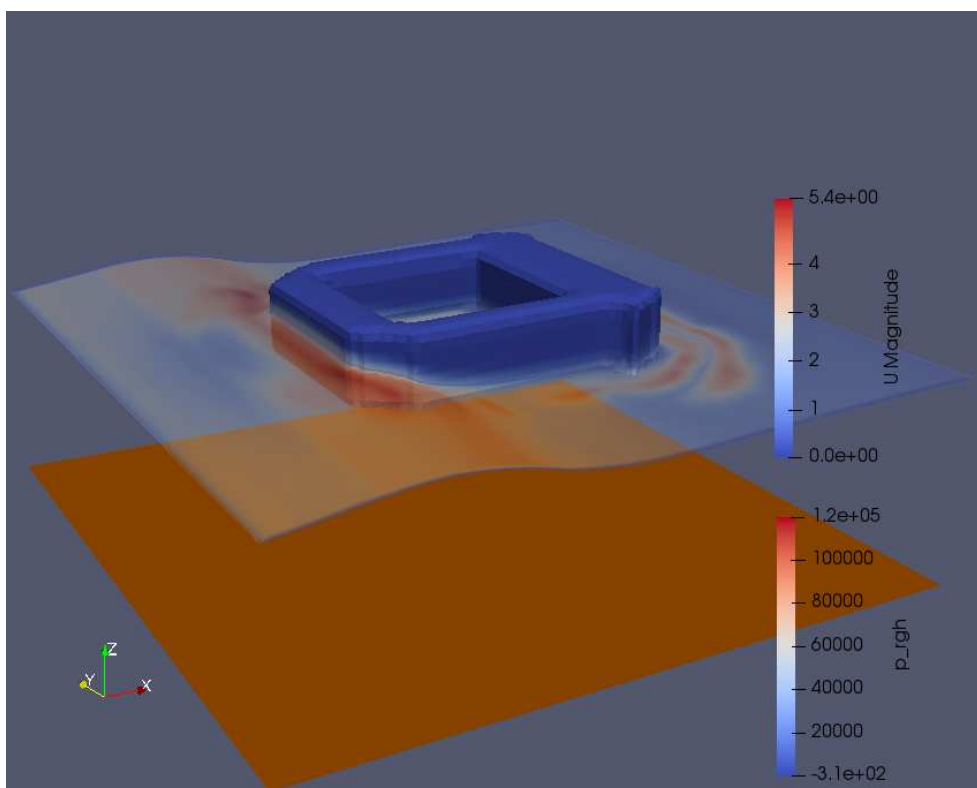


Figura 7.19: Postprocessing - pressione e velocità fluido-struttura

## 7. Modello finale

---

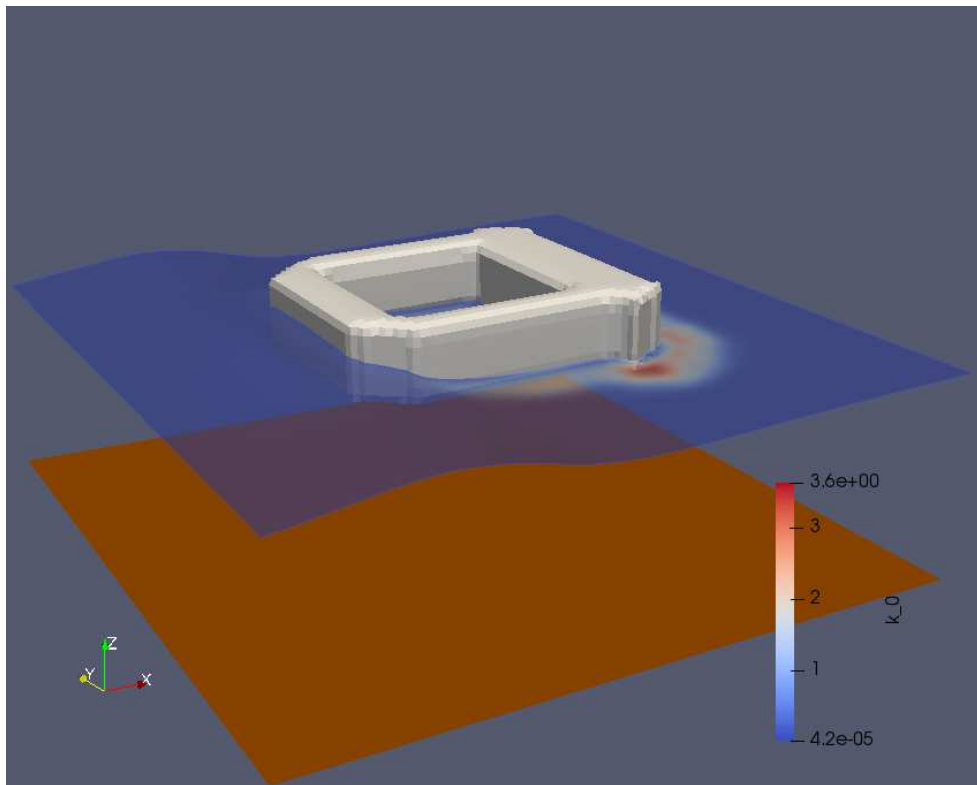


Figura 7.20: Postprocessing - energia cinetica turbolenta - snap 1

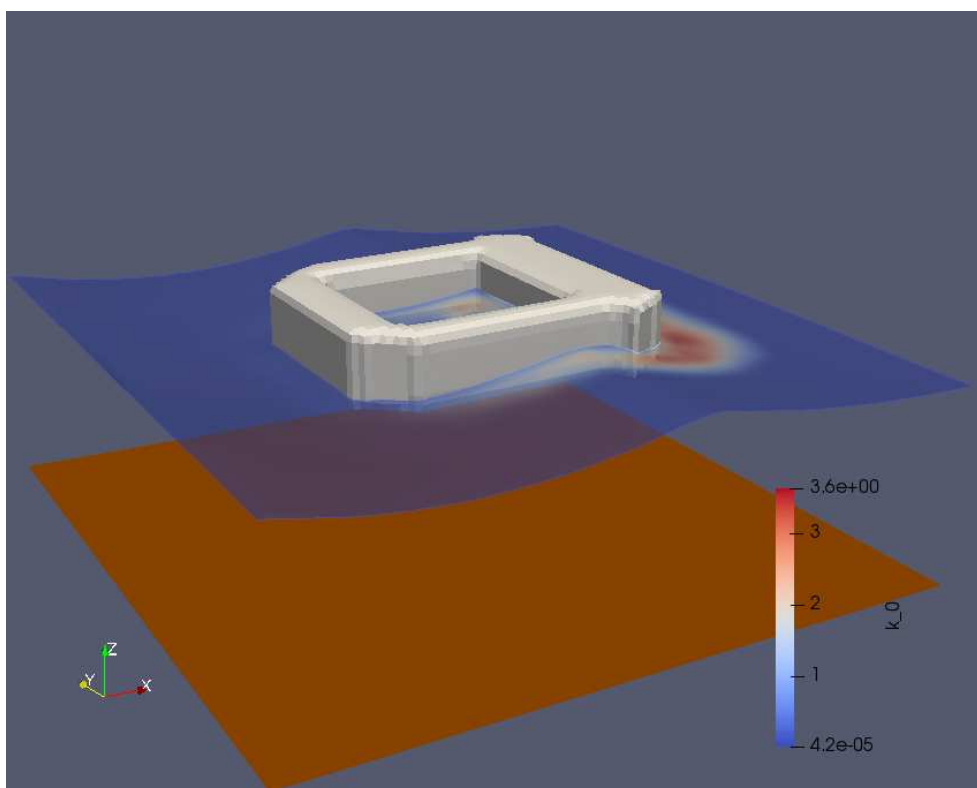


Figura 7.21: Postprocessing - energia cinetica turbolenta - snap 2

## 7. Modello finale

---

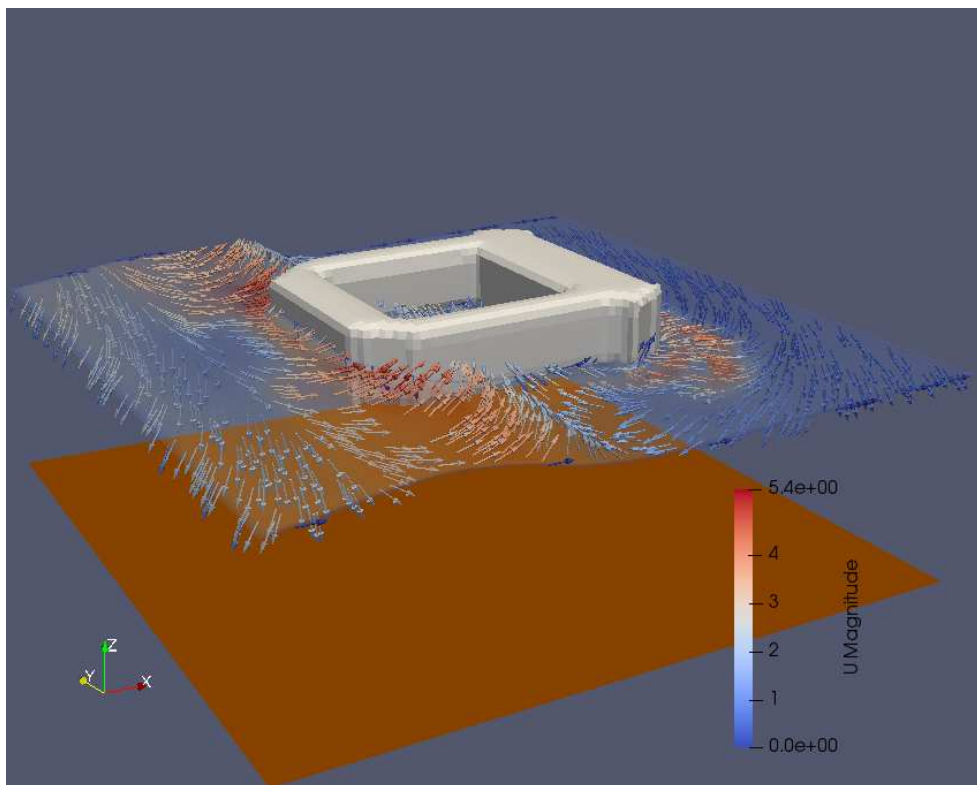


Figura 7.22: Postprocessing - vettori velocità all'interfaccia multifase - snap 1

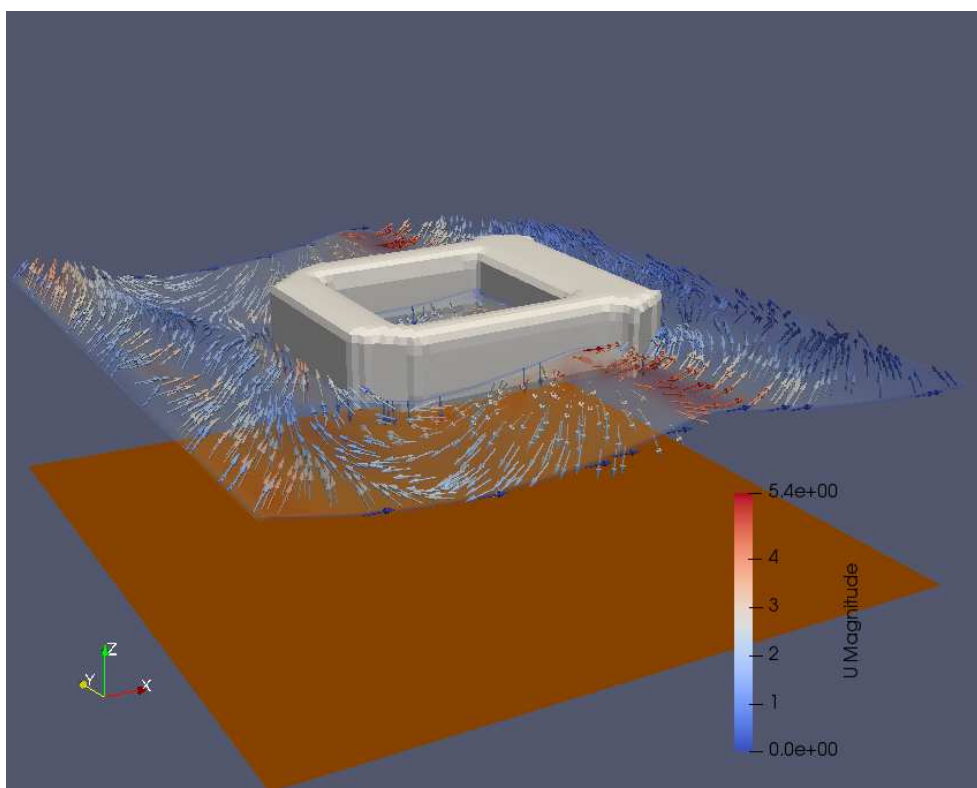


Figura 7.23: Postprocessing - vettori velocità all'interfaccia multifase - snap 2

# 8

## Conclusioni

Lo scopo di questo lavoro di tesi è stato quello di realizzare un modello in grado di simulare il comportamento di una struttura galleggiante, in particolare una Barge-based platform per l'alloggio di una turbina eolica, sotto l'effetto delle forzanti marittime, quali onde e correnti, utilizzando il software opensource OpenFOAM. Inizialmente ci si è concentrati sullo studio del software e delle basi teoriche della CFD (Computational Fluid Dynamics); una volta compresi i rudimenti di OpenFOAM, si è cercato di capire come personalizzare un modello in ogni suo aspetto, dalla mesh alle condizioni iniziali, dagli schemi risolutivi al postprocessing. Il primo modello customizzato è una semplificazione di quello finale, utile a fare pratica con il software prima di concentrarsi sul lavoro finale: il modello 2D multifase ha una geometria flottante semplice (un rettangolo), una mesh dinamica e due restraints ai vertici inferiori (molle lineari elastiche); la semplicità di quest'ultimo non ne riduce l'importanza, fin tanto che i risultati prodotti si è visto essere in completo accordo con le previsioni analitiche del caso in esame. A valle di ciò si è passati al lavoro finale, passando a una geometria complessa tridimensionale,

## 8. Conclusioni

---

una Barge-based platform, in scala 1:1; sono state fatte modifiche al programma, implementando una libreria esterna, Moody, in grado di modellare ancoraggi a catenaria e taut-moorings. Accoppiando Moody e OpenFOAM, utilizzando in alcune fasi dei parametri di default, è stato possibile far girare il modello, simulando 4 diverse casistiche e comparandole per cercare di comprendere come interagiscono fluido e struttura, e quali possono essere eventuali punti di forza e criticità di quest'ultima.

Possibili sviluppi di tale progetto prevedono un'analisi più dettagliata dei parametri del modello, come il raffinamento della mesh o l'esatta descrizione fisica degli ancoraggi. Sarà inoltre utile studiare in modo più capillare diverse configurazioni di carico, inserendo onde e correnti derivate da studi locali della zona di posizionamento della turbina. Un'ulteriore analisi più approfondita può essere effettuata simulando ancoraggi di diverso tipo (ad esempio catenarie) in diverse configurazioni, cercando di arrivare ad una soluzione ottimizzata del modello.

Un'ulteriore osservazione è di fondamentale importanza: il modello in esame prevede lo studio e la modellazione della sola interazione fluido-struttura; il moto ondoso superficiale, nelle zone a largo della costa (dove vengono posizionati i parchi eolici galleggianti), è generalmente generato da correnti ventose con velocità e direzione costanti per un determinato intervallo di tempo. Queste correnti, oltre a incresparsi la superficie del mare, generando appunto delle onde che avranno la stessa direzione di propagazione del vento, impattano sulla turbina con delle conseguenti forze di spinta sulla struttura, e infine sugli ancoraggi. La spinta aerodinamica sulla torre, sulle pale, il momento generato dalla rotazione di queste ultime e l'eventuale effetto giroscopico dato dall'inclinazione dell'asse di rotazione rispetto all'orizzontale, sono effetti trascurati nel



## 8. Conclusioni

---

modello, ma giocano in effetti un ruolo essenziale negli spostamenti del basamenti, e quindi nelle sollecitazioni sugli ancoraggi. L'aggiunta della componente aerodinamica nel modello è quindi un passaggio imperativo se si vuole studiare l'effettivo comportamento della piattaforma ai carichi ambientali.

In conclusione, è utile ricordare che un modello numerico non può sostituire un esperimento fisico: una combinazione dei due metodi può essere efficace per identificare i relativi vantaggi e svantaggi, per poi utilizzare queste tecniche in modo complementare; integrando la modellazione numerica alla sperimentazione in laboratorio, sarà possibile infatti ampliare i casi di studio, simulando differenti condizioni marittime e di ancoraggio in tempi e costi estremamente ridotti, e con una risposta di equivalente efficacia.



# Bibliografia

- [1] *1Dof spring mass damper system by Michael Alletto*. 2022. URL: [https://wiki.openfoam.com/1Dof\\_spring\\_mass\\_damper\\_system\\_by\\_Michael\\_Alletto](https://wiki.openfoam.com/1Dof_spring_mass_damper_system_by_Michael_Alletto) (cit. a p. 40).
- [2] D. Angeli e M. Cavazzuti. *OpenFOAM introduzione al software e tutorials guidati*. Modena Linux User Group, 2016 (cit. a p. 29).
- [3] D. Apsley. *Turbulence modelling*. 2023 (cit. a p. 21).
- [4] E. Pugliese Carratelli e F. Dentale. *Appunti del corso di Idraulica Marittima*. Catania: Università di Catania, 2018 (cit. a p. 11).
- [5] L. Collegiani. *Analisi, modellazione e ottimizzazione del funzionamento di un sistema di conversione per l'energia da moto ondoso*. Pisa: Università degli Studi di Pisa, 2014 (cit. a p. 8).
- [6] *Cos'è l'eolico offshore. L'energia del vento da turbine in mare aperto*. 2016. URL: <https://www.lifegate.it/eolico-offshore> (cit. a p. 1).
- [7] L. Damiani et al. *Meccanica del moto ondoso - Teoria di Airy*. Bari: Politecnico di Bari, 2021 (cit. a p. 7).
- [8] *DynamicMeshDict*. 2023. URL: <https://openfoamwiki.net/index.php/DynamicMeshDict> (cit. a p. 33).
- [9] Kailia Energia. *Progetto preliminare per la realizzazione di un parco eolico offshore*. Brindisi: Kailia Energia S.r.l., 2021 (cit. alle pp. 16, 17).

- [10] Hitachi Energy. *Eolico offshore flottante: di cosa si tratta e quali vantaggi genera*. infobuildenergia, 2021 (cit. a p. 1).
- [11] H. Jasak. *Dynamic Mesh Handling in OpenFOAM*. ResearchGate, 2009 (cit. a p. 34).
- [12] Kumar e Navdeep. *A rigidBodyDynamics tutorial with demonstrations*. In *Proceedings of CFD with OpenSource Software*. Nilsson. H., 2016 (cit. a p. 33).
- [13] *OpenFOAM User Guide. Mesh generation with the snappyHexMesh utility*. 2023. URL: <https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4.4-mesh-generation-with-the-snappyhexmesh-utility> (cit. a p. 32).
- [14] A. Paci. *Numerical analysis of floating structures for off-shore and harbour engineering*. Bologna: Alma Mater Studiorum - Università di Bologna, 2018 (cit. a p. 47).
- [15] *restraint Class Reference*. 2023. URL: [https://cpp.openfoam.org/v4/classFoam\\_1\\_1RBD\\_1\\_1restraint.html](https://cpp.openfoam.org/v4/classFoam_1_1RBD_1_1restraint.html) (cit. a p. 40).
- [16] O. Sallam. *Creating a Numerical wave tank on OpenFOAM BlueC-FD*. Department of Ocean Engineering CFD Community, 2022 (cit. a p. 43).
- [17] *Teoria lineare del moto ondoso*. 2019. URL: [https://it.wikipedia.org/wiki/Teoria\\_lineare\\_del\\_moto\\_ondoso](https://it.wikipedia.org/wiki/Teoria_lineare_del_moto_ondoso) (cit. a p. 7).
- [18] M. Urquhart. *A tutorial of the sixDofRigidBodyMotion library with multiple bodies*. Göteborg: Chalmers University of Technology, 2016 (cit. a p. 38).
- [19] *Water Waves in OpenFOAM*. 2017. URL: <https://cfd.direct/openfoam/free-software/waves/> (cit. a p. 35).
- [20] M. Wu. *Coupled motion of two floating objects*. Göteborg: Chalmers University of Technology, 2016 (cit. a p. 37).

- [21] R. Yang et al. *Current Status and Future Trends for Mooring Systems of Floating Offshore Wind Turbines*. Sustainable Marine Structures, 2022 (cit. a p. 14).