

# POLITECNICO DI TORINO

## FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN  
INGEGNERIA ASTRONAUTICA ED AEROSPAZIALE

DIPARTIMENTO DI INGEGNERIA MECCANICA ED AEROSPAZIALE



TESI DI LAUREA MAGISTRALE IN  
INGEGNERIA AEROSPAZIALE

## ANALISI PRESTAZIONALE DI UNA BARCA A VELA DA REGATA TRAMITE COMPUTATIONAL FLUID DYNAMICS CON DINAMICA A TRE GRADI DI LIBERTÀ

**Relatore:**  
**Prof.** Domenic D'Ambrosio

**Candidato:**  
Gabriele Bianco m265664

**Relatore Aziendale:**  
**Ing.** Maurizio Cossutti  
Cossutti Yacht Design

ANNO ACCADEMICO 2020/2021

# Ringraziamenti

Come un grande concerto è concretizzato da un cantante, anche il titolo di laurea è portato a termine da uno studente, capace di studiare, di imparare e sostenere gli esami. Tuttavia, come il cantante viene supportato da uno staff composto da tante persone che si occupano di tutti gli aspetti fondamentali necessari a fare sì che il cantante riesca a sostenere il concerto, anche lo studente ha bisogno di una grande squadra che lo supporti nelle operazioni essenziali per la buona riuscita dei suoi compiti; una squadra composta dalla famiglia, dagli amici, dai professori, dagli incontri "lungo il percorso".

Alla fine di questo percorso di Laurea Magistrale desidero ringraziare di cuore tutta la mia famiglia. Saper di poter contare in qualsiasi momento sul loro aiuto mi ha aiutato ad affrontare le più ardue difficoltà con fermezza e convinzione.

Un grande ringraziamento al prof. Domenic D'Ambrosio ed agli ingegneri della *C&G Yacht Design*: all'Ing. Cossutti, all'Ing. Ganz e all'Ing. Cattozzi per il supporto tecnico indispensabile al completamento della presente tesi.

Il Polito Sailing Team è il team studentesco che mi ha accolto all'inizio del percorso di laurea magistrale e mi ha dato la possibilità di prendere parte ai progetti ed alle realizzazioni di un gruppo di studenti del Politecnico di Torino estremamente appassionati e competenti; un grande ringraziamento per le bellissime esperienze vissute che mi hanno aiutato a sviluppare competenze tecniche e personali essenziali.

Ika, Luca, Peppe e Lorenzo, sono i compagni d'avventura, li ringrazio pensando alle risate, ai momenti di preoccupazione durante il percorso, ai progetti seguiti insieme, purtroppo la pandemia che ha colpito il mondo intero in quest'ultimo anno non ci permette di festeggiare insieme, ma prima o poi, quando tutto sarà finito, faremo ancora più festa.

# Abstract

Lo sviluppo delle barche da regata è un processo ambito da tutti gli ingegneri. Realizzare mezzi altamente performanti, costruiti mediante le ultime tecnologie disponibili navali richiede maggiori attenzioni nella fase progettuale.

Il progresso tecnologico ha reso disponibile una nuova metodologia di studio che può essere utilizzata anche nell'ambito della progettazione delle barche a vela; questa nuova tecnologia è la *fluidodinamica computazionale*. La fluidodinamica computazionale, tramite la soluzione di particolari equazioni, porta al calcolo delle grandezze fondamentali di un flusso di fluido e permette quindi di calcolare anche l'interazione con il fluido stesso dei corpi immersi in esso. E' facile capire come questa tecnologia possa risultare fondamentale nel calcolo delle prestazioni delle imbarcazioni.

La possibilità di osservare, all'interno del modello, la barca durante la navigazione è un aspetto di forte utilità agli ingegneri i quali, attraverso i codici CFD, possono arrivare al calcolo estremamente preciso delle condizioni di navigazione delle geometrie da loro prodotte. Tuttavia l'utilizzo dei codici di calcolo fluidodinamici nel settore velico non è, ad oggi, ancora ampiamente presente; i costi elevati di software e hardware necessari nella maggior parte dei casi risulta proibitivo alle piccole/medie imprese.

La presente tesi ha lo scopo di descrivere nei dettagli il processo di sviluppo di un modello di calcolo per analisi fluidodinamica computazionale orientato al calcolo delle prestazioni di navigazione di una barca a vela da regata di classe MELGES 24. La particolarità dello studio svolto risiede nella possibilità di calcolare lo stato di equilibrio su tre gradi di libertà della barca e, soprattutto, l'implementazione attraverso l'ambiente di calcolo open-source OpenFoam e la realizzazione delle simulazioni su dispositivi Hardware di tipo home che rendono l'intero processo relativamente economico. Tramite il confronto con i dati ottenuti mediante il VPP e mediante i dati misurati durante l'utilizzo della barca dalle diverse istituzioni che ne hanno fatto uso nel

tempo, si giunge a conclusione che i modelli CFD sviluppati portano ad una conoscenza dei dati di assetto della barca di notevole precisione. L'elaborato si conclude con la descrizione dettagliata dei dati ottenuti mediante le simulazioni costruite attraverso il software open-source Paraview.

# Contents

<b>Ringraziamenti</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>Introduzione</b>	<b>10</b>
<b>1 Barca a vela da regata</b>	<b>11</b>
1.1 Componenti principali . . . . .	11
1.2 La classe velica . . . . .	12
1.2.1 La classe Melges 24 . . . . .	13
1.3 Le regate veliche . . . . .	14
<b>2 Il progetto della barca a vela</b>	<b>15</b>
2.1 Funzionamento della barca a vela . . . . .	15
2.1.1 Il triangolo del vento . . . . .	16
2.2 Terminologia e sistemi di riferimento . . . . .	16
2.2.1 Andature . . . . .	17
2.3 Forze applicate . . . . .	19
2.3.1 Resistenza della carena . . . . .	21
2.3.2 Le forze laterali, la deriva ed il timone . . . . .	26
2.4 Il VPP, Velocity Prediction Program . . . . .	27
2.5 Introduzione della CFD nel progetto della barca a vela . . . . .	28
<b>3 Computational Fluid Dynamics</b>	<b>30</b>
3.1 Le equazioni della fluidodinamica . . . . .	30
3.2 Soluzione delle equazioni . . . . .	32
3.2.1 Discretizzazione delle equazioni attraverso i volumi finiti	33
3.2.2 Boundary Conditions . . . . .	33
3.3 Approccio di soluzione al caso incomprimibile, bifase, e in-	
stazionario . . . . .	35
3.3.1 Le equazioni per il fluido incomprimibile . . . . .	35

3.3.2	L'algoritmo SIMPLE . . . . .	35
3.3.3	L'algoritmo PISO . . . . .	39
3.3.4	L'algoritmo PIMPLE . . . . .	41
3.3.5	Il caso bifase e l'approccio VOF e l'approccio Eulerian multiphase . . . . .	42
3.4	La turbolenza . . . . .	44
3.5	Movimento della mesh . . . . .	47
3.5.1	L'approccio Overset Mesh . . . . .	48
3.6	Struttura macroscopica di un codice CFD . . . . .	50
3.7	Convergenza ed analisi della simulazione . . . . .	51
3.7.1	Validazione del modello . . . . .	52
<b>4</b>	<b>Open Foam, Paraview e la struttura della simulazione</b>	<b>54</b>
4.1	OpenFoam, il software opensource per la Computational Fluid Dynamics . . . . .	55
4.1.1	Utilities . . . . .	56
4.1.2	I solutori . . . . .	56
4.1.3	Le librerie . . . . .	57
4.1.4	Le Dictionaries . . . . .	57
4.2	La simulazione CFD implementata su OpenFoam . . . . .	58
4.2.1	Il codice InterFoam e overInterDyMFoam . . . . .	59
4.3	Paraview . . . . .	63
<b>5</b>	<b>Caso studio</b>	<b>64</b>
5.1	Obbiettivi della simulazione CFD . . . . .	65
5.2	Struttura della simulazione . . . . .	67
5.2.1	Dominio di calcolo . . . . .	68
5.2.2	Costruzione e dimensionamento della mesh . . . . .	70
5.3	Condizioni al contorno . . . . .	75
5.3.1	Velocità, $U$ . . . . .	76
5.3.2	Pressione specifica, $p_{rgh}$ . . . . .	77
5.3.3	ZONE ID . . . . .	77
5.3.4	PointDisplacement . . . . .	78
5.3.5	Energia cinetica turbolenta, $k$ . . . . .	78
5.3.6	Rateo di dissipazione della turbolenza, $\epsilon$ . . . . .	79
5.3.7	Turbolent viscosity, $\tilde{\nu}$ . . . . .	79
5.4	Il codice principale: controlDict, fvSolution, fvScheme . . . . .	80
5.4.1	controlDict . . . . .	80
5.4.2	fvSolution . . . . .	80
5.4.3	fvSchemes . . . . .	82
5.5	Dinamica della carena, <i>dynamicMeshDict</i> . . . . .	82

5.5.1	Schema di funzionamento della barca nella simulazione	86
5.6	Modello di Turbolenza	87
5.7	Le altre Dictionaries utilizzate	88
5.7.1	surfaceFeatureExtractDict	88
5.7.2	topoSetDict	88
5.7.3	refineMeshDict	89
5.7.4	setFieldsDict	89
5.7.5	decomposeParDict e reconstructParDict	89
5.8	Gestione del codice e lancio della simulazione	89
5.9	Unione degli effetti	91
5.10	Modifica delle simulazioni per il regime di navigazione sim- metrico	92
5.11	Hardware disponibile per lo svolgimento delle simulazioni	94
<b>6</b>	<b>Analisi dei risultati</b>	<b>96</b>
6.1	Convergenza ed analisi della simulazione	96
6.2	Analisi sui dati numerici della simulazione	97
6.2.1	Numero di Courant	97
6.2.2	Ventilazione numerica	99
6.2.3	Wall $y^+$	100
6.3	Analisi dei parametri fisici ottenuti dalle simulazioni	101
6.3.1	Caratterizzazione della curva di resistenza della carena	102
6.3.2	Analisi del regime di navigazione e del sistema di onde generato dalla barca	106
6.4	Utilizzo dei dati nell'ambito del progetto	116
<b>7</b>	<b>Conclusioni e progetti futuri</b>	<b>117</b>

# List of Figures

1.1	Classi veliche olimpiche [15]	13
1.2	Melges 24 [1]	13
2.1	Triangolo del vento[11]	16
2.2	Assi corpo barca	16
2.3	Andature della barca a vela [2]	18
2.4	Schema delle forze agenti sulla barca a vela [11]	20
2.5	Sistema di onde generato dalla barca [3]	22
2.6	Interferenza costruttiva e distruttiva tra l'onda di prua e l'onda di poppa [11]	24
2.7	Curva di resistenza d'onda [11]	25
2.8	Polare ottenuta mediante il Velocity Prediction Program [6]	28
3.1	Discretizzazione ai volumi finiti	33
3.2	Schema algoritmo SIMPLE [9]	38
3.3	Schema algoritmo PISO [9]	40
3.4	Schema algoritmo PIMPLE	41
4.1	OpenFoam logo [8]	55
4.2	Pseudo-algoritmo OverInterDyMFoam [4]	62
4.3	Interfaccia di Paraview	63
5.1	Dominio del codice di calcolo	69
5.3	Discretizzazione dominio spaziale	72
5.4	Discretizzazione del Prism Layer sulla superficie dello scafo	74
5.5	Schema delle forze esterne inserite nella dinamica della carena	87
5.6	Dominio del codice di calcolo	93
6.1	Distribuzione numero di Courant ottenuto nelle simulazioni	98
6.2	Distribuzione numero di Courant ottenuto nelle simulazioni	100
6.3	Distribuzione numero di Courant ottenuto nelle simulazioni	101
6.4	Analisi del regime simmetrico calcolato dalle simulazioni CFD	103

6.5	Scena sistema d'onda in regime dislocante con $V=3$ m/s . . .	104
6.6	Scena sistema d'onda in regime planante con $V=4.5$ m/s . . .	104
6.7	Evoluzione del sistema d'onde generato dalla barca . . . . .	105
6.8	Risultati dell'andatura di traverso . . . . .	107
6.9	Risultati dell'andatura di bolina . . . . .	108
6.10	Risultati dell'andatura di gran lasco . . . . .	109
6.11	Andamento grafico dei risultati ottenuti mediante la CFD e il VPP . . . . .	111
6.12	Distribuzione di pressione sulla carena in andatura . . . . .	113
6.14	Composizione della resistenza complessiva sviluppata dalla carena	115

# List of Tables

5.1	Dimensioni del dominio . . . . .	68
5.2	Dimensione della mesh di background . . . . .	70
5.3	Dimensione della mesh di overset . . . . .	70
5.4	Condizioni al contorno per la velocità . . . . .	76
5.5	Condizioni al contorno per la pressione specifica . . . . .	77
5.6	Condizione al contorno per l'indice della zona, ZONE ID . . . . .	77
5.7	Condizioni al contorno per il <i>Point Displacement</i> . . . . .	78
5.8	Condizioni al contorno per l'energia cinetica turbolenta . . . . .	78
5.9	Condizioni al contorno per il rateo di dissipazione della turbolenza . . . . .	79
5.10	Condizioni al contorno per il <i>Point Displacement</i> . . . . .	79
5.12	Tolleranza sui residui per le simulazioni della carena . . . . .	80
5.11	Ingressi della dictionary <i>controlDict</i> . . . . .	81
5.13	Parametri per il controllo della soluzione dell'equazione della Volume Fraction per lo svolgimento delle simulazioni della carena . . . . .	81
5.14	Parametri per il controllo dell'algoritmo PIMPLE . . . . .	82
5.15	Parametri numerici di settaggio del codice per la dinamica del corpo in movimento . . . . .	83
6.1	Risultati numerici delle simulazioni in regime simmetrico . . . . .	102
6.2	Errore dei risultati numerici delle simulazioni in regime simmetrico . . . . .	102
6.3	Velocità calcolate dal VPP in input alla simulazione per l'andatura di traverso . . . . .	107
6.4	Dati in input alla simulazione per l'andatura di bolina . . . . .	108
6.5	Dati in input alla simulazione per l'andatura di gran lasco . . . . .	109
6.6	Risultati numerici della barca completa . . . . .	110
6.7	Errore numerico tra le simulazioni CFD e i modelli VPP . . . . .	110
6.8	Composizione della Resistenza all'avanzamento . . . . .	114

# Introduzione

Sin dall'antichità l'uomo è stato coinvolto nella ricerca delle modalità migliori per rendere possibili ed accelerare gli spostamenti tra città e popolazioni ed agevolare il trasporto di merci tra le città.

Inizialmente, l'approccio più efficace è stato quello di sfruttare la natura nei suoi eventi.

Nacque così la navigazione dei corsi d'acqua terrestri e, successivamente, anche dei mari più estesi. Vennero costruiti i primi scafi propulsi dalla resistenza aerodinamica di estese porzioni di tessuto; vengono costruite i primi prototipi di barca a vela.

La prima imbarcazione a vela risale al 6000 a.C., successivamente, Egizi, Greci, Fenici ed Arabi cominciarono a farne un uso più consistente e documentato, con lo scopo di commerciare, compiere spostamenti per esplorazioni e per scopi bellici tramite le rotte marine.

Numerose sono le modifiche che hanno interessato le barche a vela negli anni, diverse forme di vele, diverse tipologie di albero, fino ad arrivare alle imbarcazioni da competizione attuali, molto leggere e performanti, capaci di sfruttare anche il più leggero dei venti.

Attualmente, le imbarcazioni a vela non vengono più impiegate per scopi commerciali, bensì vengono utilizzate per attività ludiche; si organizzano gare anche molto importanti tra imbarcazioni a vela da regata. Progettare una barca a vela oggi, significa creare un mezzo che sia veloce, facile da guidare, sicuro e che possa adattarsi alle disverse condizioni di vento.

La seguente tesi ha come obiettivo lo studio tramite Fluidodinamica Computazionale del funzionamento di una barca a vela di classe MELGES 24.

# Chapter 1

## Barca a vela da regata

*Dalla passione dell'uomo per la navigazione a vela, questa attività diventa un modo per l'uomo di spingersi oltre. Attraverso le regate veliche, gli equipaggi con le proprie imbarcazioni, studiate e modificate nel minimo dettaglio, si sfidano sui circuiti di gara per stabilire la barca e l'equipaggio migliore.*

*Andare molto veloci sfruttando il più leggero dei venti è il sogno e l'obiettivo che spinge gli ingegneri a sviluppare progetti sempre più complessi, per una serie di tipologie di imbarcazioni anche molto diverse tra di loro.*

*In questo capitolo vengono introdotti i concetti di base inerenti alla barca a vela e ad uno dei modi di utilizzarla: le regate veliche.*

### 1.1 Componenti principali

La barca a vela è un mezzo costituito da tre macro-gruppi, che hanno interfaccia con uno o con entrambi i fluidi che intervengono durante la navigazione, sono suddivisi per funzione:

- La **carena**, è la parte che galleggia nell'acqua, è quindi immersa in parte in acqua ed in parte in aria, serve a contenere e trasportare il carico pagante;
- Le **vele**, sono i "propulsori" di ogni barca a vela. Sfruttano le forze aerodinamiche agenti su di esse per produrre e trasferire allo scafo le forze per il movimento tramite l'albero. In funzione del tipo di imbarcazione, della classe di regata sono installate sulla barca una o più vele di tipologie diverse;

- Le **appendici**, sono corpi idrodinamici che, messi in velocità rispetto al fluido in cui sono immersi servono a produrre forze per il controllo e la stabilità della barca a vela. Le due appendici sempre presenti sono il timone, per il controllo del *leeway angle* e la deriva, studiata per agire sulla dinamica del piano orizzontale, per limitare lo scarroccio. Nell'ultimo periodo, gli studi per la riduzione della resistenza dell'imbarcazione hanno portato all'aggiunta di ulteriori appendici che hanno lo scopo di produrre una forza che si oppone al peso e riesce a far alzare la barca sull'acqua, staccando lo scafo dalla fase liquida. Così facendo è possibile ridurre drasticamente la resistenza viscosa dello scafo stesso, queste appendici vengono chiamate *foils*.

Le parti di una barca a vela descritte sopra sono presenti su qualsiasi barca a vela ed hanno un'importanza nell'ambito della fluidodinamica, nella sua componente idrodinamica o aerodinamica. Come si può facilmente supporre non sono gli unici presenti su una barca a vela da un punto di vista globale.

Numerose parti vengono installate con funzioni strutturali, per le barche più complesse anche sistemi elettronici per la misurazione di dati di funzionamento e di navigazione della barca. Inoltre, per poter pilotare la barca in maniera controllata, in funzione della classe velica, vengono installati tutti i sistemi di interfaccia e controllo.

## 1.2 La classe velica

Le imbarcazioni da regata vengono costruite con l'obbiettivo di partecipare a delle gare che, per le barche a vela vengono chiamate regate. Nei numerosi sport che coinvolgono dei mezzi, automobilismo motociclismo ecc., per partecipare ad una determinata gara è fondamentale distinguere la tipologia del mezzo e garantire delle regole precise per la specifica categoria. Nell'ambito della vela, l'insieme di regolamenti che coinvolgono sia le modalità di costruzione sia di armo della barca sono contenuti nella *classe velica*. La classe velica, quindi, regola tutte le caratteristiche della barca affinché sia idonea a partecipare alle regate riservate alla sua categoria.

Esistono, attualmente, una moltitudine di classi veliche; le classi olimpiche per le derive, per esempio sono: Finn, 49er, 470, Laser, Laser Radial. Per quanto riguarda le chiglie, ossia categorie per le quali lo sbandamento è limitato da contrappesi, esistono altre classi di spessore internazionale, tra queste troviamo: Fun, Meteor, Star e Melges24, per esempio.

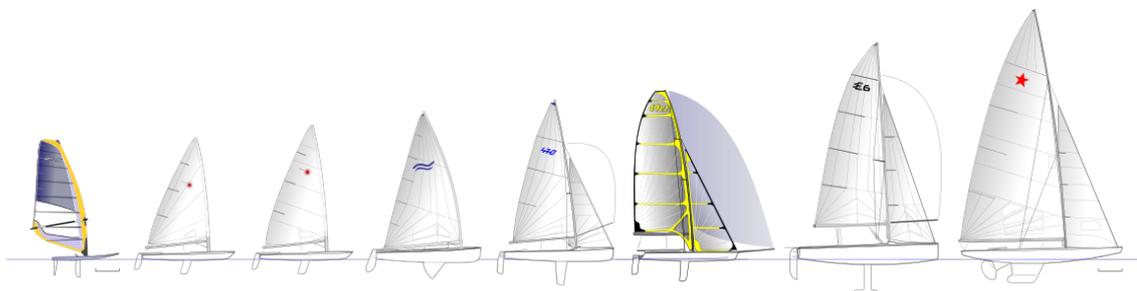


Figure 1.1: Classi veliche olimpiche [15]

Il soggetto della seguente tesi è l'imbarcazione di classe MELGES 24, approfondita nel seguito.

### 1.2.1 La classe Melges 24

La classe Melges 24 è un'imbarcazione costruita per la prima volta in America dalla casa navale *Melges* e progettata da *John Reichel* e *Jim Pugh*. La classe Melges 24 è una classe monotipo, pertanto, le caratteristiche tecniche sono rigidamente descritte e verificate al fine di poter classificare l'imbarcazione e poter partecipare alle regate di categoria.



Figure 1.2: Melges 24 [1]

Il Melges 24 è una barcha a vela a chiglia, quindi è provvista di un *bulbo* all'estremità della deriva per generare il momento di rollio adatto a limitare, in condizioni di equilibrio, l'angolo di rotazione sull'asse longitudinale. La classe in esame è un'imbarcazione monoscafo la cui carena è lunga  $7m$  con geometrie pensate per facilitare il regime planante con una limitata curvatura della carena nella sezione di poppa. È uno scafo molto leggero, realizzato in fibra di vetro. L'equipaggio è costituito da minimo 3 e massimo 5 membri.

Il MELGES 24 è propulso da un piano velico (la configurazione delle vele) che consiste in fiocco, randa e gennaker. Quest'ultimo adatto ai regimi con il vento in poppa. Il piano velico totale ha una superficie di  $93m^2$  considerando che il gennaker misura  $62m^2$ , randa e fiocco misurano circa  $31m^2$

Il peso complessivo della barca è di  $808kg$  mentre il peso massimo dell'equipaggio è limitato, per ragioni di manovrabilità, a  $375kg$ .

### 1.3 Le regate veliche

Una regata velica è una competizione che vede coinvolte le barche a vela; per ogni classe sono disponibili specifiche tipologie di regate.

L'oggetto delle regate, per una determinata classe, consiste nella percorrenza di un percorso prestabilito nel minor tempo possibile, nell'ambito del quale la barca si può trovare in condizioni di navigazione anche differenti. Come in tutti gli sport in cui gareggiano dei mezzi comandati dall'uomo, la vittoria può derivare sia dall'essere umano che la pilota, sia dalla qualità della macchina.

Le manovre vanno eseguite alla perfezione, ma il progetto può prevedere una barca più facilmente manovrabile o, in caso contrario, una barca instabile e difficile da comandare. Portare una barca a vela ad alte velocità, a parità di vento, dipende sia dall'abilità dell'equipaggio ma anche dal livello di perfezione del progetto.

Il circuito da svolgere, spesso chiamato nel gergo come *campo gara*, può essere un tracciato delimitato da delle boe apposite come per le regate in lago, oppure può essere un percorso definito da un punto di partenza ed un punto di arrivo come per le grandi regate in stile Ocean Race.

Alle regate veliche possono partecipare una o più categorie di barche a vela, distinte per caratteristiche di progetto e criteri di funzionamento, le sopraccitate classi veliche.

## Chapter 2

# Il progetto della barca a vela

*Trasformare un'idea in un oggetto reale e funzionante è l'obbiettivo di un progetto. Per sviluppare un progetto efficiente ed efficace è necessario adattare le teorie di base, consolidate in anni di studi, all'idea ed alle esigenze personali, al fine di realizzare un'opera che svolga le funzioni previste nella maniera pensata.*

*In questo capitolo vengono descritte le teorie utili al progetto di massima della barca a vela; in particolare si concentra l'attenzione sullo scafo e sulla resistenza sviluppata da esso durante la navigazione e sull'analisi prestazionale della barca completa.*

### 2.1 Funzionamento della barca a vela

Il funzionamento della barca a vela dipende dal vento. Il vento che interagisce con le vele produce una forza che viene trasferita allo scafo che, come reazione, si muove. Il punto di partenza per studiare il funzionamento di ogni barca a vela, quindi, è il flusso d'aria presente nell'ambiente di navigazione, questo fenomeno naturale può manifestarsi in diverse forme e l'imbarcazione dev'essere progettata per riuscire a sfruttare al meglio tutte queste modalità in cui si manifestano le correnti d'aria. In ambito navale, le caratteristiche della navigazione in funzione del vento, vengono chiamate *andature* che si contraddistinguono per il diverso *vento apparente* utilizzabile, le diverse forze agenti sull'intero sistema e il diverso assetto che la barca assume.

### 2.1.1 Il triangolo del vento

Nell'ambito dello studio del funzionamento delle vele è fondamentale il *triangolo del vento*. Il triangolo del vento è un'entità matematica che serve a calcolare il vettore del vento apparente, ovvero la velocità, in modulo e verso, del flusso d'aria relativo alle vele. Questa componente rappresenta la velocità relativa tra aria e vele e quindi il vettore necessario per il calcolo della portanza sviluppata dalle vele stesse.

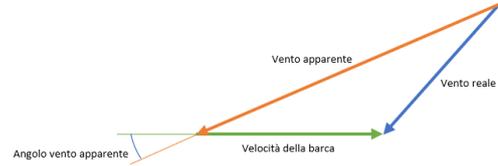


Figure 2.1: Triangolo del vento[11]

Il triangolo del vento consiste nella composizione vettoriale delle seguenti velocità in gioco:

- **Vettore vento:** corrisponde al vettore *vento reale* in figura 2.1. Il vettore rappresenta la velocità, in modulo direzione e verso del vento, rispetto ad un sistema di riferimento fisso e standard;
- **Vettore barca:** Corrispondente alla *velocità della barca* in figura 2.1. È la velocità della barca contenente sia lo scarroccio che l'avanzamento;

La somma dei due vettori descritti genera un terzo vettore, il vento apparente, in figura 2.1 indicato con il vettore *vento apparente*. Il vento apparente, è il vento su cui si svolgono le considerazioni sul funzionamento delle vele.

## 2.2 Terminologia e sistemi di riferimento

Per l'analisi del sistema barca a vela si fa riferimento agli assi di una terna cartesiana centrata nel baricentro della barca. Gli assi corpo della barca sono:

- **asse X, longitudinale**, la cui direzione è allineata alla congiungente poppa-prua ed il verso è positivo a prua;
- **asse Y trasversale**, ortogonale all'asse longitudinale sul piano orizzontale della barca;
- **Asse Z verticale**, perpendicolare agli altri assi precedentemente descritti e contenuto nel piano di simmetria della barca.

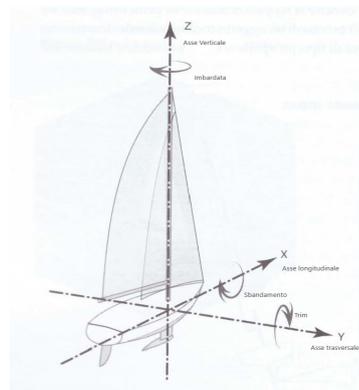


Figure 2.2: Assi corpo barca

Durante la navigazione il vettore velocità non è allineato all'asse longitudinale per la presenza della componente laterale della forza aerodinamica generata dalle vele; la velocità è contenuta nel piano orizzontale con una componente allineata all'asse longitudinale ed una componente nella direzione dell'asse trasversale, quest'ultima, è chiamata velocità di **scarroccio** e determina l'entità dello spostamento laterale durante l'avanzamento della barca.

Nelle condizioni di equilibrio, la risultante delle forze agenti sui componenti della barca, genera il momento sull'asse trasversale che determina la rotazione della barca a formare l'angolo di **trim**. Inoltre, le forze idonee alla limitazione dello scarroccio provocano la rotazione sull'asse longitudinale che viene chiamata **sbandamento**.

L'angolo tra la direzione dell'asse longitudinale e il vettore velocità della barca viene indicato con il nome di **leewayangle** ed è importante nella valutazione del triangolo del vento in quanto descrive la posizione relativa delle vele rispetto al vento. In funzione dell'andatura, l'equipaggio gestisce questo angolo agendo sul timone.

### 2.2.1 Andature

Le andature, particolari configurazioni caratteristiche di navigazione, vengono raggruppate in maro gruppi e sono valide per tutte le classi. Le caratteristiche delle andature sono le seguenti:

- **Bolina**. Il verso di avanzamento è opposto al verso del vento, in gergo si dice che la barca risale il vento. Mantenendo questa andatura le vele sono disposte in maniera tale da generare portanza e trasferire alla barca una forza trainante che permette appunto di risalire il vento. La bolina può essere "stretta" o "larga" in funzione dell'angolo, più ci si avvicina ai  $90^\circ$  di angolo relativo più si allarga la bolina. Più l'angolo è stretto ed è vicino al limite dell'angolo morto, più la bolina è stretta.
- **Traverso**. È l'andatura durante la quale le vele sono poste a circa  $90^\circ$  rispetto al vento. È l'andatura potenzialmente più veloce.  
Oltre questo valore di angolo iniziano le andature a favore di vento in cui la barca viene spinta dal vento.
- **Lasco** e **Gran lasco**. Quando il vento relativo alle vele ha un angolo compreso rispettivamente tra  $100^\circ$  e  $130^\circ$  per il lasco e  $130^\circ$  e  $170^\circ$  per

il granlasco. Durante la navigazione su queste andature la forza sulle vele è più allineata alla direzione della barca e quindi lo scarroccio è più limitato. Di conseguenza, anche il momento di rollio è più limitato rispetto alle altre andature dove le vele generano portanza e la componente trasversale è notevole.

- **Poppa.** In questa andatura, il vento è "dritto in poppa", si sfrutta pienamente la resistenza aerodinamica delle vele. La barca è disposta parallelamente al vento nella direzione concorde ad esso.
- **Angolo morto.** In questa condizione, l'angolo tra le vele ed il vento apparente è troppo basso per far funzionare correttamente le superfici veliche che risultano "sgonfiate" ed inefficaci.

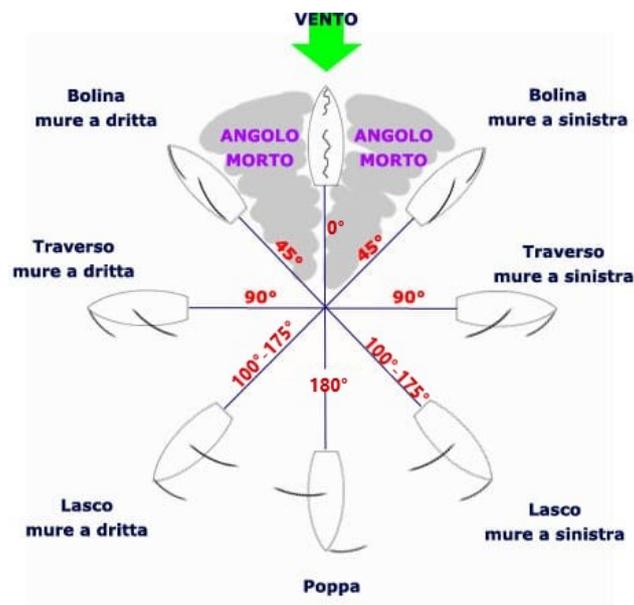


Figure 2.3: Andature della barca a vela [2]

## 2.3 Forze applicate

Durante il funzionamento della barca a vela, indipendentemente dall'andatura, diverse sono le forze applicate ai vari componenti e che devono essere tenute in considerazione durante il progetto sia per una valutazione delle prestazioni sia per garantire l'integrità dell'imbarcazione.

Considerando l'imbarcazione in condizioni di equilibrio le forze applicate rispetto al sistema di riferimento *assi vento* sono:

- Sul **piano orizzontale** (piano  $x-y$ ) nella direzione di avanzamento:
  - **Resistenza fluidodinamica**, è la forza che si oppone al moto della barca, è generata sia dalla carena che dalle appendici e dalle vele. Il vettore resistenza è allineato all'asse vento longitudinale in verso opposto al vettore velocità.
  - **Driving force**, è la forza che "traina" la barca. Essendo la componente utile all'avanzamento è l'entità più importante che si deve ottimizzare per avere un mezzo veloce. È anch'essa parallela all'asse vento longitudinale in verso concorde al vettore velocità;
- Sul **piano orizzontale** nella direzione normale rispetto alla direzione di avanzamento:
  - **Forza aerodinamica laterale**, allineata all'asse vento trasversale, tende a fare scarrocciare la barca, è la componente di forza che deve essere maggiormente limitata per ottimizzare la direzione di avanzamento della barca;
  - **Forza idronamica laterale** è necessaria a bilanciare la forza aerodinamica laterale per limitare la velocità di scarroccio, l'accoppiamento delle due forze laterali produce un momento di rollio che produce l'angolo di sbandamento. Il momento di rollio generato da queste due forze dev'essere bilanciato dal bulbo all'estremità della deriva, che fa da contrappeso, e dall'equipaggio che sposta il baricentro della barca;
- Nel **piano verticale** (piano  $x-z$ ) :
  - **Peso della barca**, applicato nel baricentro e perpendicolare alla superficie terrestre;

- **Forza di galleggiamento**, anch'essa in direzione verticale, si oppone al peso della barca, generata dalla *spinta di Archimede*. Il punto di applicazione dipende dalla geometria della carena. La distanza del punto di applicazione dal baricentro della carena genera un ulteriore momento di rollio della barca che si oppone allo sbandamento;
- Componente **aerodinamica ed idrodinamica per sbandamento**. La presenza di un angolo di sbandamento provoca l'inclinazione fuori dal piano delle forze laterali sul piano orizzontale. La conseguente scomposizione della forza laterale sia aerodinamica che idrodinamica in una direzione verticale e orizzontale produce la componente verticale delle forze laterali sia aerodinamica che idrodinamica.

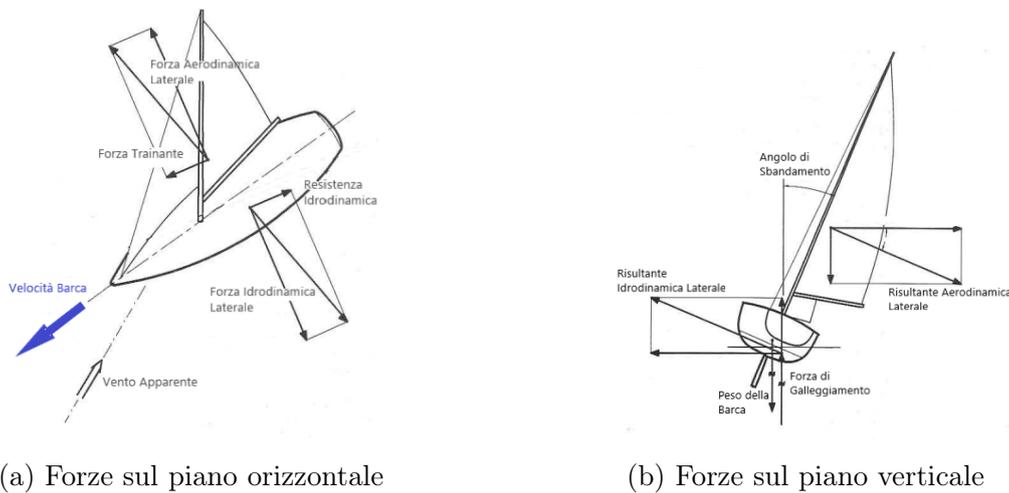


Figure 2.4: Schema delle forze agenti sulla barca a vela [11]

Lo scafo, posto in movimento dalla forza trainante generata dalle vele e trasferita ad esso mediante l'albero e le sartie è sottoposto alle forze idrodinamiche. La forza principale è la resistenza idrodinamica; con le sue componenti, è la forza che si oppone al moto della barca ed è dovuta al fatto che lo scafo è posto ad una velocità relativa con l'acqua. Se è presente anche la forza laterale, e nella maggior parte dei casi è sempre presente, questa causa il cosiddetto scarroccio ossia il movimento laterale dello scafo. Analogamente all'avanzamento si genera una forza idrodinamica laterale sullo scafo. Insieme a queste due forze è presente la forza di galleggiamento che sostiene la barca intera ed è data dalla legge di Archimede per cui un corpo immerso in un

fluido è sottoposto ad una forza che si oppone alla forza di gravità di modulo pari al peso del fluido spostato, che nell'ambito dello studio dei liquidi assume valori importanti data la densità. Nel seguito vengono esplicitate le componenti di resistenza che agiscono sulla carena.

### 2.3.1 Resistenza della carena

La resistenza è la forza che si oppone all'avanzamento ed è la forza che è necessario ridurre il più possibile per ottenere un mezzo più efficiente; un corpo immerso in un fluido sarà sempre sottoposto ad una resistenza derivante dagli sforzi d'attrito, tangenziali alla superficie del corpo e dalla pressione, normale alla superficie, le quali componenti a loro volta sono parallele ed opposte al moto; inoltre, possono essere presenti altre cause di resistenza. Per lo scafo di un'imbarcazione a vela la resistenza è dovuta ai seguenti effetti:

- Resistenza d'attrito;
- Resistenza viscosa di pressione;
- Resistenza d'onda.

#### Resistenza d'attrito

La resistenza per attrito viscoso è la componente preponderante della resistenza all'avanzamento globale dello scafo, data l'elevata viscosità e l'elevata densità dell'acqua, sia essa dolce che salata. La resistenza viscosa nasce dal fenomeno dell'attrito viscoso, per il quale le prime molecole a diretto contatto con la superficie della carena si fermano sulla superficie, è presente la *no slip condition*, e si sviluppa lo strato limite.

La resistenza d'attrito viscoso, dipende da parametri come la viscosità dinamica del fluido, dalla rugosità della superficie e dalla velocità del flusso.

#### La resistenza d'onda

La componente di resistenza d'onda rappresenta un fonte di resistenza idrodinamica importante e che va tenuta in considerazione sin dalle prime fasi progettuali. Per capire com'è generata la resistenza d'onda è utile osservare il sistema ondoso a valle dello scafo che, come un corpo che cade in acqua, genera un sistema ondoso che propaga sulla superficie del fluido nel tempo e nello spazio.

"A differenza di un corpo che cade in acqua, generando una serie di cerchi concentrici con centro il punto di caduta del corpo, nel caso di un'imbarcazione, la sorgente, è in movimento e il sistema ondoso presenta delle caratteristiche descritte dalla teoria del *sistema ondoso di Kelvin*" [11] .

Assumendo la barca come oggetto puntiforme in movimento nell'acqua si nota come il sistema di onde generato dal punto stesso sia formato da due tipologie di onde ben distinte: il primo tipo è l'**onda divergente**, che propaga lateralmente al corpo e mano a mano si allontana dall'asse di mezzeria. Il secondo tipo è l'**onda trasversale**, che è un'onda che si sviluppa nella direzione normale al moto del corpo e collega le due estremità dell'onda divergente.



Figure 2.5: Sistema di onde generato dalla barca [3]

Abbandonando l'ipotesi di sorgente puntiforme, e quindi avvicinandosi al sistema barca, si può capire come avviene la generazione dell'insieme di onde a valle descritto sopra e come si manifesta la resistenza d'onda. I punti fondamentali che intervengono nella generazione delle onde sono i punti ad alta pressione, collocati nelle zone di prua e a poppa della barca. Si può vedere in figura 2.6 come sia dalla prua che dalla poppa si sviluppano sia le onde divergenti che trasversali. Non solo, per gestire la resistenza d'onda è fondamentale studiare l'interazione dell'onda di prua e di poppa. L'onda, rappresentata dalla variazione dell'altezza della superficie di contatto tra aria ed acqua nasce per gli sbalzi di pressione che avvengono per la variazione di geometria descritta sopra. Una decompressione genera un abbassamento della superficie, mentre una compressione consegue in un innalzamento del pelo libero. Le variazioni di pressione dovute alla geometria, sono le prime cause della generazione dell'onda e quindi della resistenza d'onda.

Passando all'analisi in termini matematici del sistema ondoso nelle immediate vicinanze della barca durante il suo funzionamento, si può affermare che l'onda propaga con la stessa velocità della barca.

Da questa prima considerazione e dalla conoscenza delle caratteristiche della carena, in particolare la  $L_{WL}$ , allora si può calcolare il numero di creste dell'onda entro lo scafo e quindi il tipo di interazione che si ha tra l'onda di prua, che si sviluppa lungo lo scafo, e l'onda di poppa che nasce a poppa. Come si può osservare dallo schema in figura, in funzione della velocità e della lunghezza bagnata della barca ci possono essere diversi gradi di interferenza.

Le grandezze coinvolte nella fenomenologia sono le seguenti:

- Lunghezza bagnata,  $L_{LW}$ ;
- Velocità della barca (e quindi velocità di propagazione dell'onda),  $V = \sqrt{\frac{g \cdot \lambda}{2\pi}}$ ;
- Gravità,  $g = 9.81 \frac{m}{s^2}$ .

Svolgendo un'analisi dimensionale per trovare un parametro adimensionale che regoli queste tre grandezze si ottiene il numero di Froude descritto dalla relazione:

$$F_n = \frac{V}{g \cdot L_{LW}} \quad (2.1)$$

Il numero di Froude descrive l'interazione tra l'onda di prua e l'onda di poppa e rappresenta, per il settore navale, un analogo del numero di Reynolds per l'aerodinamica.

Passando all'analisi dei diversi regimi di navigazione si osserva lo schema in figura 2.6. Osservando il primo caso, l'onda di prua si sviluppa e la velocità e lunghezza della carena sono tali che la cresta dell'onda di prua coincide con la cresta dell'onda di poppa. Questo fenomeno è un fenomeno di interferenza costruttiva e l'onda complessiva che lascia la barca è amplificata.

La seconda casistica presenta una fenomenologia opposta. In questo caso la lunghezza della barca è sempre la stessa ma cambia la sua velocità. La lunghezza d'onda dell'onda di prua a sua volta è diversa e quindi si sviluppa un'onda la cui cresta non coincide più con la cresta dell'onda di poppa. In questo caso si ha interferenza distruttiva e il sistema ondoso è smorzato.

Tramite il *Froude Number*, quindi, si è in grado di capire quante onde si sviluppano lungo la lunghezza bagnata. Si hanno le seguenti casistiche teoriche:

$F_n < 0.40$  Lungo l'onda di prua si propaga manifestando più creste una delle quali andrà ad interferire, in maniera costruttiva o distruttiva, con la cresta dell'onda di poppa;

$F_n = 0.40$  Le condizioni sono tali che solo un'onda attraversa la lunghezza bagnata dello scafo;

$Fn > 0.40$  In questa condizione la lunghezza d'onda è più ampia della lunghezza bagnata della carena e quindi solo una porzione dell'onda attraversa lo scafo. Questa è una condizione particolare in quanto determina il punto in cui l'assetto si modifica in maniera consistente e l'angolo di trim aumenta, la carena entra in regime planante.

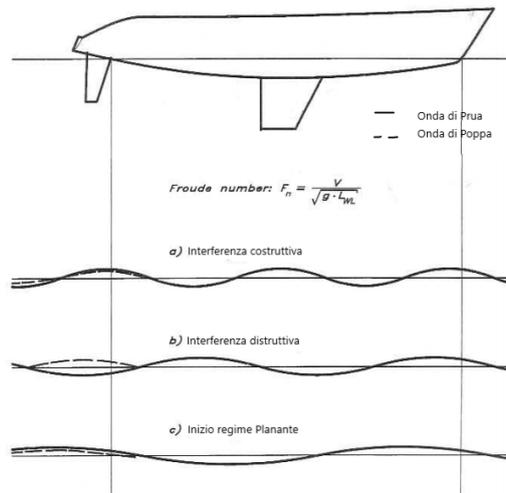


Figure 2.6: Interferenza costruttiva e distruttiva tra l'onda di prua e l'onda di poppa [11]

Per lo studio della resistenza d'onda quindi molta importanza ricopre il parametro del numero di Freud, infatti, come in molti casi accade negli studi fluidnamici, si fa ricorso ai parametri adimensionali piuttosto che alle grandezze. Nel caso specifico il numero di Freud viene preso come primo riferimento per quantificare la resistenza d'onda generata da un'imbarcazione durante la navigazione. Il grafico in figura 2.8 mostra l'andamento della resistenza al variare del numero di Freud. Si può osservare l'andamento monotono crescente, ma attenzione va posta sulle variazioni di pendenza della curva.

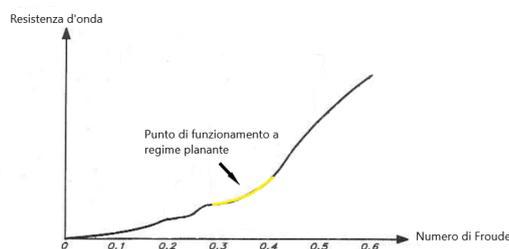


Figure 2.7: Curva di resistenza d'onda [11]

In alcune zone, infatti si ha un comportamento che provoca un aumento repentino della resistenza al variare del Froude, quindi per una data carena, al variare della sua velocità. Questi punti, durante la navigazione vanno evitati se si vuole avere un mezzo più prestante. Durante la progettazione, quindi, decisivo è il calcolo della curva per una data imbarcazione per prevedere al meglio i punti svantaggiosi per le performance della barca.

Osservando il grafico, il salto più importante, si ha, in linea teorica, in corrispondenza del  $Froude = 0.4$  in questo punto raggiungere la zona con aumento più graduale della resistenza d'onda è più difficile, data l'elevata velocità necessaria, ma è il punto di interesse per le imbarcazioni che devono funzionare in regime planante. Una barca che riesce a raggiungere in anticipo rispetto ad altri progetti questo regime allora è una barca più performante.

Nella realtà, il punto in cui avviene il passaggio a regime planante non avviene perfettamente in corrispondenza del numero di Froude descritto dalla teoria, bensì ad un valore più alto; questo fenomeno è giustificato dal fatto che le eventuali sporgenze a prua ed a poppa rappresentano un'aumento della lunghezza bagnata nominale.

### La resistenza di pressione viscosa

Questa componente di resistenza è data dai gradienti di pressione presenti sulla lunghezza dello scafo. A prua la pressione è maggiore, vicino alla prua è presente il punto di stagnazione e quindi in corrispondenza di questo punto la pressione raggiunge il valore di pressione d'arresto, punto in cui la pressione è più alta. Segue una zona di espansione mano a mano che la carena aumenta la larghezza nella direzione trasversale, analogamente ad un profilo alare, ed una successiva ricomprensione, maggiore nel caso in cui a poppa avviene la separazione. Questo fenomeno è dato dal fatto che la geometria di una carena di una barca a vela è analoga a quella di un profilo alare, benché le sue funzioni sono diverse.

Partendo dalla prua, il punto di minimo spessore, la geometria va mano a mano allargandosi, sia sul piano orizzontale che sul piano verticale nella parte immersa in acqua. Questa variazione di geometria provoca un aumento della pressione statica del fluido.

Il punto di massimo spessore si raggiunge dopo il punto medio sulla lunghezza della carena dopo il quale la geometria si restringe con una conseguente espansione del fluido. La zona di poppa costituisce un punto critico, in questa zona, dove la geometria viene "troncata" è presente il distacco della vena fluida.

### 2.3.2 Le forze laterali, la deriva ed il timone

Deriva e timone, le appendici di tutte le barche a vela, sono delle superfici portanti che interagiscono con la fase liquida del fluido bifase globale che interessa la navigazione. Queste appendici sono del tutto paragonabili alle ali degli aerei, infatti, la loro sezione è rappresentata da un profilo alare simmetrico ottimizzato per l'idrodinamica, mentre la forma in pianta può essere di diverso tipo, in funzione della complessità del progetto e delle prestazioni richieste.

Timone e deriva sviluppano portanza, che analizzata nel sistema barca rappresenta una forza laterale, dall'analisi dell'andatura della barca a vela, infatti, emerge la necessità di bilanciare la componente laterale delle forze aerodinamiche, sviluppate dalle vele.

Come descritto nel paragrafo 2.3 le vele sviluppano una componente di forza normale all'asse vento longitudinale che tende a far traslare lateralmente la barca, in gergo "scarrocciare". Per limitare il fenomeno le barche sono equipaggiate di deriva, una superficie portante che, in presenza di scarroccio, genera la forza idrodinamica orizzontale. Maggiore è lo scarroccio, maggiore è l'angolo di incidenza del flusso d'acqua rispetto alla deriva, a parità di *leeway angle*, e maggiore è la forza che si oppone alla forza laterale delle vele. Il punto di applicazione della forza laterale della deriva si trova al disotto del baricentro della barca, e la distanza rappresenta il braccio del momento di sbandamento generato della deriva. Inoltre, anche la forza laterale aerodinamica, con applicazione al disopra del baricentro della barca, genera un momento di sbandamento concorde a quello della deriva. La limitazione dello scarroccio, quindi, può portare ad una rotazione notevole della barca lungo l'asse longitudinale, per questo motivo, le derive delle barche di dimensione più grandi sono equipaggiate con delle zavorre, all'estremità bassa, la cui forza peso e la distanza dal baricentro sviluppa un momento adatto a

limitare lo sbandamento. Inoltre, anche l'equipaggio, durante le andature in cui le forze laterali sono più forti, si sposta per portare il baricentro in una posizione in cui la forza peso si oppone maggiormente allo sbandamento.

Il timone è la seconda appendice presente su tutte le imbarcazioni sia a vela che motorizzate. Il timone, è una superficie portante la cui portanza, corrispondente anch'essa ad una forza laterale, vista nel sistema barca, è ottenuta dal flusso d'acqua dato dal moto della barca e dall'angolo d'incidenza complessivo di angolo di scarroccio e angolo di inclinazione impostato dall'equipaggio per il controllo della barca. Il timone, infatti, genera una coppia che fa ruotare la barca rispetto all'asse verticale con lo scopo di direzionare la prua e modificare l'angolo di *leeway*. In maniera minore rispetto alla deriva, anche il timone contribuisce ad equilibrare le forze orizzontali.

## 2.4 Il VPP, Velocity Prediction Program

Lo strumento attualmente utilizzato in ambito navale per il calcolo delle prestazioni della barca è il *Velocity Prediction Program*. L'obiettivo è quello di riuscire a misurare la velocità finale della barca date le condizioni di vento in ingresso e caratteristiche costruttive della barca. Insieme agli output prestazionali il VPP riesce a calcolare anche gli assetti di equilibrio che assume lo scafo alle determinate andature.

Il *velocity prediction program* è basato sulla soluzione delle equazioni della dinamica della barca e successiva soluzione di equazioni ottenute su base empirica che calcolano le forze agenti sulla carea, in particolare la resistenza, e riescono a valutare anche la velocità complessiva data una determinata condizione di vento.

La soluzione del VPP permette di ricavare, quindi, i dati necessari a disegnare la polare di navigazione, che è un grafico nel quale vengono plottati i vettori velocità della barca per ogni vento apparente di navigazione. Osservando la figura 2.8, ogni linea rappresenta il modulo della velocità vera del vento mentre l'angolo indicato sui "raggi" del grafico indica la direzione del vento apparente. Il vettore congiungente la curva con il centro rappresenta la velocità della barca che si ottiene in quelle condizioni di vento.

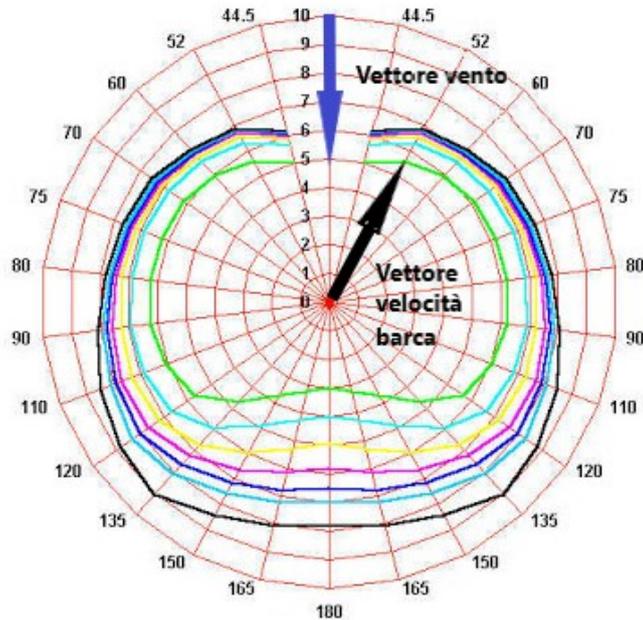


Figure 2.8: Polare ottenuta mediante il Velocity Prediction Program [6]

## 2.5 Introduzione della CFD nel progetto della barca a vela

La computational Fluid Dynamics è un mezzo, come verrà descritto in maniera approfondita nei successivi capitoli, per risolvere le equazioni della fluidodinamica per via numerica e quindi permette di risolvere un determinato flusso calcolandone pressioni, velocità e se richiesto anche le temperature.

Nel settore navale grande impiego può assumere questo approccio per la progettazione ed ottimizzazione delle imbarcazioni a vela. Tutti i componenti principali già elencati possono essere studiati tramite la *computational fluid dynamics* con grande precisione ed affidabilità. L'obiettivo è quello di superare le imprecisioni dei metodi empirici, come il VPP, e poter descrivere nel complesso e dettagliatamente il flusso attorno alla barca per poterne studiare "da vicino" tutte le caratteristiche; forma dell'onda, regime di navigazione ecc... .

La nota a svantaggio, per la quale la CFD è ancora oggi una tecnologia non ampiamente utilizzata nel settore navale, resta il costo; hardware e software necessari per qualità e tempistiche di calcolo soddisfacenti spesso sono i re-

sponsabili di un aumento notevole dei costi; per questa ragione, solo nei casi in cui è necessaria una precisione ed un'ottimizzazione "spinta", come per le barche da regata avanzate, questi metodi vengono presi in considerazione trascurando la variabile costo.

Inoltre, anche le difficoltà nell'utilizzo della Computational Fluid Dynamics sono elevate, il background teorico e l'esperienza necessaria per poter applicare correttamente i codici di calcolo fluidodinamici deve essere abbastanza consolidati e specifici.

Attualmente vengono però sperimentati numerosi metodi per risolvere il problema del costo della CFD, in maniera tale da poterla utilizzare agilmente anche nell'ambito commerciale per sviluppare barche sempre migliori. Nei prossimi capitoli verrà descritto un approccio di sviluppo delle simulazioni CFD tale da assicurare un contenimento dei costi e dei risultati che giustifichino la spesa.

# Chapter 3

## Computational Fluid Dynamics

*Studiare l'effetto di un fluido che interagisce con un corpo immerso in esso è una capacità fondamentale nell'ambito ingegneristico che permette di riuscire ad ottimizzare un progetto e raggiungere la piena consapevolezza dei fenomeni che intervengono durante l'interazione tra il corpo e il fluido che lo circonda. Diversi sono i metodi che permettono di studiare nello specifico un fenomeno fluidodinamico, le due tecniche più adottate fino ad ora sono tecniche sperimentali e numeriche. Le prime hanno l'obiettivo di simulare in un laboratorio la situazione di funzionamento più vicina alla condizione reale per effettuare, durante il funzionamento stesso, le misurazioni necessarie. I metodi numerici, invece, mirano al calcolo delle caratteristiche del fluido attraverso la soluzione delle equazioni fondamentali dei fluidi, le equazioni di Navier Stokes, risolte con complessi algoritmi svolti da potenti calcolatori.*

*Il seguente capitolo è dedicato alla descrizione dei procedimenti teorici di calcolo utilizzati per svolgere le analisi utili al caso della navigazione delle barche. Le modifiche del sistema di equazioni della fluidodinamica sviluppate per il caso multifase, incomprimibile e turbolento verranno analizzate. Le formule riportate nel presente capitolo sono tratte dalle seguenti fonti [10][9][5]*

### 3.1 Le equazioni della fluidodinamica

Il punto di partenza per lo studio della fluidodinamica computazionale sono le equazioni fondamentali della fluidodinamica, anche chiamata di Navier-Stokes. Questo set di equazioni è ottenuto tramite l'interpretazione dei concetti teorici di *conservazione della massa*, *seconda legge di Newton* e *conservazione dell'energia* sviluppati in termini matematici per il caso di una

particella fluida; un'entità abbastanza piccola da poter descrivere in maniera circa puntuale le caratteristiche del fluido nel dominio di studio ma abbastanza grande da poter contenere al suo interno il numero di molecole che garantisca un numero di *Knudsen* per la condizione di *continuo* e che garantisca la possibilità di utilizzare le grandezze termodinamiche medie all'interno della particella stessa.

Nel seguito vengono descritte le prime due equazioni fondamentali; la terza, relativa alla conservazione dell'energia, non verrà utilizzata nel seguito perciò il suo approfondimento viene trascurato. Le equazioni vengono descritte sia nella forma integrale, sia nella forma differenziale dal punto di vista Euleriano (volume di controllo fisso nello spazio e flusso che lo attraversa).

### Legge di Conservazione della massa

La prima equazione della fluidodinamica rappresenta una riscrittura del principio di conservazione della massa adattata al flusso di particelle fluide. Il concetto alla base é:

*La variazione di massa all'interno di un volume di controllo è pari al flusso di massa attraverso le superfici del volume di controllo stesso (ipotizzando che all'interno del volume di controllo non sono presenti sorgenti di massa).*

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \vec{v} \cdot \vec{n} dS &= 0 \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) &= 0 \end{aligned} \quad (3.1)$$

### Legge di bilancio della quantità di moto

L'equazione di bilancio della quantità di moto si basa sulla seconda legge di Newton:

$$\text{Forza} = \text{massa} \cdot \text{accelerazione} = m \cdot \frac{dv}{dt}$$

*Le forze agenti su una particella fluida causano la variazione della sua quantità di moto.*

Le forze agenti sulla particella possono essere forze di superficie (o di contatto), pressione e sforzi viscosi, e di volume (a distanza), peso che spesso viene trascurato ed altre forze che dipendono dal caso studio.

Formulata per un flusso di particelle fluide diventa:

$$\frac{\partial}{\partial t} \int_V \rho \vec{v} dV + \int_S \rho \vec{v} \vec{v} \cdot \vec{n} dS + \int_S p \vec{I} \cdot \vec{n} dS - \int_S \vec{\tau} \cdot \vec{n} dS = \int_V \rho \vec{f} dV \quad (3.2)$$

$$\frac{\partial(\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \vec{v}) + \nabla p - \nabla \cdot \vec{\tau} = \rho \vec{f}$$

## 3.2 Soluzione delle equazioni

Lo studio del fluido consiste nella soluzione delle equazioni descritte, che permette di ottenere le componenti di velocità e la pressione. Nei casi in cui è presente anche l'equazione dell'energia la soluzione prevede il calcolo delle grandezze termodinamiche che intervengono, come la temperatura o l'entalpia.

Risolvere numericamente il sistema di equazioni integro differenziali significa risolvere un sistema algebrico di equazioni nelle sue incognite, ottenuto mediante discretizzazione delle equazioni integro-differenziali con l'obiettivo di calcolare i valori delle grandezze incognite in istanti temporali e zone spaziali discreti.

La discretizzazione può avvenire seguendo diversi schemi, differenze finite, volumi finiti o elementi finiti ed è necessaria dal momento che intervengono i cosiddetti metodi numerici per la soluzione. Nell'ambito della fluidodinamica e coerentemente con gli argomenti e le metodologie attuate nel presente elaborato, nel seguito, viene esposto il metodo di discretizzazione attraverso i volumi finiti.

### 3.2.1 Discretizzazione delle equazioni attraverso i volumi finiti

Il metodo ai volumi finiti prevede la suddivisione del dominio spaziale in *celle* rappresentate da un certo volume, che nel caso più semplice possono essere dei parallelepipedi, mentre in altri casi possono essere formati da geometrie poliedriche più complesse. Durante la soluzione per ognuno dei volumi che formano il dominio intero, vengono risolte le equazioni e quindi vengono calcolate le grandezze incognite, successivamente mediate all'interno del singolo volume di controllo.

Il dominio di calcolo assume quindi le seguenti caratteristiche, è formato da:

- Facce tra le celle, sulle quali vengono calcolati i flussi in entrata ed uscita;
- Punti di calcolo, sui quali vengono mediate le grandezze all'interno della cella;
- Superficie di contorno, è l'insieme di superfici delle singole celle che sono posizionate sul bordo del dominio, sono estremamente importanti per l'impostazione delle *boundary conditions* essenziali per la soluzione delle equazioni.

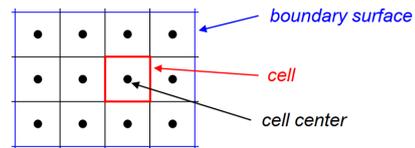


Figure 3.1: Discretizzazione ai volumi finiti

La generazione dei volumi, chiamati in letteratura *CVs*, *Control Volumes* può avvenire seguendo due logiche: la tecnica più utilizzata permette di suddividere il dominio in celle ed assegnare il punto computazionale al centro cella. Si può svolgere la suddivisione del dominio anche nella maniera inversa, assegnando prima il punto computazionale e successivamente costruire le facce attorno ad esso.

### 3.2.2 Boundary Conditions

In una simulazione fluidodinamica, un ruolo fondamentale è ricoperto dalle condizioni al contorno. In letteratura vengono indicate come *boundary conditions* e servono a completare il sistema di equazioni dei modelli scelti per

la simulazione fornendo dei valori delle grandezze del fluido note in determinate zone del dominio di calcolo. Inoltre, nelle simulazioni con discretizzazione temporale, anche i valori di campo all'istante iniziale devono essere implementati e definiti attraverso il processo di *inizializzazione* delle variabili.

Le tipologie di base per la definizione delle condizioni al contorno sono due: la condizione di *Neumann*, che assegna un valore noto al gradiente o alla divergenza della grandezza in corrispondenza del contorno e la condizione e la condizione di *Dirichlet* che, invece, assegna un valore noto alla grandezza al contorno.

Partendo da questi due concetti di base la formulazione delle condizioni al contorno può variare ed essere personalizzata in funzione del caso studio. Esistono delle categorie di condizioni al contorno standard che vengono implementate nella maggior parte dei software. Le condizioni al contorno più utilizzate sono le seguenti:

- **inlet**, rappresentano l'ingresso del fluido. Questa condizione impone un determinato valore di velocità alla superficie;
- **outlet**, permette la fuoriuscita del fluido dal dominio, in questa boundary viene imposto un certo valore di pressione del flusso;
- **wall**, è la condizione di parete. Può essere settata con la condizione di *slip* o *no-slip* condition;
- **symmetry**, rappresenta una condizione di "specchio" rispetto alla superficie di contorno.

Altre condizioni al contorno sono implementate all'interno dei codici di calcolo e consistono in evoluzioni delle condizioni di base elencate sopra.

### 3.3 Approccio di soluzione al caso incomprimibile, bifase, e instazionario

#### 3.3.1 Le equazioni per il fluido incomprimibile

Nel caso di flusso incomprimibile, per definizione di incomprimibilità, la densità è fissata ed una variazione di pressione non ha effetto sulla densità stessa. In termini matematici:

$$\frac{\partial \rho}{\partial p} = 0 \quad (3.3)$$

Questa caratteristica porta al disaccoppiamento dell'equazione dell'energia, che può essere risolta a posteriori e non è necessaria alla soluzione del sistema composto dalle prime due equazioni.

Inoltre, l'indipendenza della densità dalla pressione, provoca una "separazione" tra l'equazione della quantità di moto, e l'equazione della continuità. Un campo di velocità ottenuto con un certo campo di pressioni non è detto che soddisfi anche l'equazione della continuità, essendo che all'interno di quest'ultima non c'è nessun legame con la pressione.

Come verrà descritto nel seguito, esistono degli algoritmi che permettono di ottenere la soluzione del problema delle equazioni in un caso incomprimibile. Il nuovo set di equazioni, modificate per il caso incomprimibile, e trascurando l'equazione dell'energia è il seguente:

$$\begin{aligned} \nabla \cdot \vec{v} &= 0 \\ \rho \frac{\partial \vec{v}}{\partial t} + \rho \nabla \cdot (\vec{v}\vec{v}) + \nabla p - \nabla \cdot \vec{\tau} &= \rho \vec{f} \end{aligned} \quad (3.4)$$

La soluzione del problema per il fluido incomprimibile si ottiene mediante diversi algoritmi che sono stati sviluppati ed applicati. Il concetto di base su cui si fonda la soluzione delle equazioni di un flusso incomprimibile è il *guess* iniziale di un campo di pressioni errato e la sua correzione mediante il campo di velocità che soddisfa anche l'equazione della continuità. Nei prossimi paragrafi verranno esposti degli algoritmi consolidati e utilizzati in numerosi codici CFD che utilizzano il concetto descritto per la soluzione di questa tipologia di flussi.

#### 3.3.2 L'algoritmo SIMPLE

Il primo algoritmo studiato per la soluzione dei fluidi incomprimibili è l'algoritmo *Semi-Implicit Method for Pressure-Linked Equations*, ottenuto da Patankar

e Spalding nel 1972.

Il primo passaggio dell'algoritmo è la definizione di un campo di pressioni iniziale  $p^*$ , valore dal quale si ottiene un campo di velocità dall'equazione della quantità di moto discretizzata, tramite la relazione:

$$\begin{aligned} b_{i,j}u_{i,j}^* &= \sum b_{n,b}u_{n,b}^* + (p_{I-1,J}^* - p_{I,J}^*)T_{i,j} + c_{i,j} \\ b_{I,j}v_{I,j}^* &= \sum b_{n,b}v_{n,b}^* + (p_{I,J-1}^* - p_{I,J}^*)T_{I,j} + c_{I,j} \end{aligned} \quad (3.5)$$

Le grandezze  $u^*$  e  $v^*$  rappresentano il campo di velocità ottenuto mediante la pressione iniziale di *guess*  $p^*$ . Il campo di velocità trovato, però, non è detto che soddisfi l'equazione della continuità, non essendoci correlazione tra pressione e densità.

Si introducono, quindi, dei termini correttivi sia per le velocità che per le pressioni con lo scopo di, tramite l'equazione della continuità, correggere i campi delle grandezze fisiche in gioco. In termini matematici:

$$\begin{aligned} p &= p^* + p' \\ u &= u^* + u' \\ v &= v^* + v' \end{aligned} \quad (3.6)$$

Sostituendo i termini corretti nelle equazioni discretizzate della quantità di moto si ottiene il campo di moto esatto:

$$\begin{aligned} b_{i,j}(u_{i,j} + u_{i,j}^*) &= \sum b_{n,b}(u_{n,b} - u_{n,b}^*) + [(p_{I-1,j} - p_{I-1,j}^*) - (p_{I,j} - p_{I,j}^*)]T_{i,j} + c \\ b_{I,j}(v_{I,j} + v_{I,j}^*) &= \sum b_{n,b}(v_{n,b} - v_{n,b}^*) + [(p_{I,J-1} - p_{I,J-1}^*) - (p_{I,J} - p_{I,J}^*)]T_{I,j} + c \end{aligned} \quad (3.7)$$

Per esplicitare i termini correttivi, che rappresentano le incognite del problema, si sottraggono alle 3.7 le 3.16 ottenendo la seguente equazione:

$$\begin{aligned} b_{i,j}u'_{i,j} &= \sum b_{n,b}u'_{n,b} + (p'_{I-1,J} - p'_{I,J})T_{i,j} \\ b_{I,j}v'_{I,j} &= \sum b_{n,b}v'_{n,b} + (p'_{I,J-1} - p'_{I,J})T_{I,j} \end{aligned} \quad (3.8)$$

Caratteristica dell'algoritmo SIMPLE è l'eliminazione del termine  $\sum b_{n,b}u'_{n,b}$ . La convergenza del calcolo comporta la diminuzione del termine correttivo sia della velocità che della pressione. Infatti, la convergenza, impone che i campi di velocità e pressione finali siano quelli "corretti". Nella realtà, tuttavia, a causa dell'approccio numerico e delle approssimazioni che si svolgono il termine correttivo manterrà sempre un determinato valore diverso da zero,

seppur molto piccolo.

Le velocità, comprensive del termine correttivo diventano: Ottenuti i termini correttivi, si possono esplicitare i termini completi descritti dalle 3.7, tenendo in considerazione la semplificazione svolta prima. Si ottengono le seguenti relazioni:

$$\begin{aligned} u &= u_{i,J}^* + s_{i,J}(p'_{I-1,J} - p'_{I,J}) \\ v &= v_{I,j}^* + s_{i,J}(p'_{I,J-1} - p'_{I,J}) \end{aligned} \quad (3.9)$$

A questo punto entra in gioco l'equazione della continuità necessaria al calcolo dei termini correttivi della pressione. Questo passaggio rappresenta l'ultima fase logica per la soluzione del problema. Calcolando i termini di pressione è possibile anche calcolare il campo di velocità sostituendo questi ultimi nelle equazioni 3.9.

Nell'equazione della continuità discretizzata, si sostituiscono i termini di velocità corretti descritti dalle 3.9:

$$b_{I,J}p'_{I,J} = b_{I+1,J}p'_{I+1,J} + b_{I-1,J}p'_{I-1,J} + b_{I,J+1}p'_{I,J+1} + b_{I,J-1}p'_{I,J-1} + c'_{I,J} \quad (3.10)$$

Questa equazione rappresenta l'equazione che permette di ottenere il link tra il campo di pressioni e l'equazione della continuità nel caso di fluido incomprimibile.

I termini che compaiono nell'equazione, come moltiplicatori delle pressioni corrette si ottengono durante la discretizzazione e sono rappresentati dal prodotto tra l'area della superficie della cella e la densità:  $b_{k,z} = \rho dA_{k,z}$  e  $c'_{I,J} = (\rho u^* A)_{i,J} - (\rho u^* A)_{i+1,J} + (\rho v^* A)_{I,j} - (\rho v^* A)_{I,j+1}$ .

Il procedimento descritto sopra è di tipo iterativo e inizialmente sviluppato per casi studio stazionari, in cui il sistema non varia nel tempo. L'evoluzione del metodo SIMPLE permette di aggiungere il termine contenente la discretizzazione temporale  $\frac{\partial \rho}{\partial t}$ , quest'ultimo viene aggiunto alle equazioni descritte e compare in forma discretizzata nel termine  $\frac{(\rho_P - \rho_P^0)}{\Delta t} \Delta V$  come addendo nell'equazione di conservazione della quantità di moto. Si ottiene la seguente equazione della pressione:

$$b_{I,J}p'_{I,J} = b_{I+1,J}p'_{I+1,J} + b_{I-1,J}p'_{I-1,J} + b_{I,J+1}p'_{I,J+1} + b_{I,J-1}p'_{I,J-1} + c'_{I,J} \quad (3.11)$$

dove i coefficienti  $a_{X,Y}$  mantengono lo stesso significato rispetto al caso precedente mentre il termine  $b_{I,J}$  viene provvisto del contributo temporale come segue:

$$c'_{I,J} = (\rho u^* A)_{i,J} - (\rho u^* A)_{i+1,J} + (\rho v^* A)_{I,j} - (\rho v^* A)_{I,j+1} + \frac{(\rho_P^0 - \rho_P) \Delta V}{\Delta t} \quad (3.12)$$

In una situazione instazionaria, quindi, il processo caratterizzante l'algoritmo SIMPLE viene svolto iterativamente per ogni *time-step*.

Schematicamente, l'algoritmo SIMPLE è rappresentato nella figura sottostante 3.2.



Figure 3.2: Schema algoritmo SIMPLE [9]

### 3.3.3 L'algoritmo PISO

L'algoritmo *Pressure Implicit with Splitting Operators*, sviluppato da Issa nel 1986, è un algoritmo di tipo inizialmente non iterativo nato per il calcolo di flussi compressibili instazionari ed adattato per la soluzione di problemi stazionari ed instazionari. Il concetto alla base di questo metodo risolutivo è l'identificazione di un campo di pressioni iniziale non corretto, esattamente come nel caso del metodo precedente, ed il calcolo delle velocità con il campo di pressione iniziale come nell'algoritmo SIMPLE. La sostanziale differenza con l'algoritmo SIMPLE consiste nella presenza di due passaggi di correzione del campo di pressione.

L'algoritmo è costituito dai seguenti passaggi.

1. Il primo passaggio viene svolto in maniera analoga all'algoritmo SIMPLE, si risolve l'equazione della quantità di moto con un campo di pressioni intermedio, non corretto;
2. Il passo successivo consiste in una prima correzione della pressione e dei campi di velocità, anche in questo caso seguendo la logica dell'algoritmo SIMPLE si ottengono le equazioni per la velocità 3.9;
3. La principale differenza consiste nell'ultimo passaggio, un ulteriore passaggio correttivo. Per l'algoritmo PISO si riscrivono i campi di velocità e pressione come:

$$\begin{aligned}
 p &= p^{***} = p^* + p' + p'' = p^{**} + p'' \\
 u &= u^{***} = u^* + u' + u'' = u^{**} + u'' \\
 v &= v^{***} = v^* + v' + v'' = v^{**} + v''
 \end{aligned} \tag{3.13}$$

Dove l'apice singolo rappresenta il termine di pressione ottenuto dalla soluzione della prima equazione della pressione, mentre il doppio apice rappresenta il campo di pressioni finale ottenuto con la soluzione del secondo ciclo correttivo. Riscrivendo le 3.16 per le velocità di guess di seconda fase si ottiene:

$$\begin{aligned}
 b_{i,j}u_{i,j}^{***} &= \sum b_{n,b}u_{n,b}^{**} + (p_{I-1,j}^{***} - p_{I,J}^{***})T_{i,J} + c_{i,J} \\
 b_{I,j}v_{I,j}^{***} &= \sum b_{n,b}v_{n,b}^{**} + (p_{I,J-1}^{***} - p_{I,J}^{***})T_{i,j} + c_{I,j}
 \end{aligned} \tag{3.14}$$

Come svolto precedentemente quindi si sostituiscono le velocità  $V^{***}$  nell'equazione di bilancio della massa per ottenere un'equazione con incognita la seconda pressione correttiva  $p''$ . L'equazione ultima per il calcolo della pressione è:

$$b_{I,J}p''_{I,J} = b_{I+1,J}p''_{I+1,J} + b_{I-1,J}p''_{I-1,J} + b_{I,J-1}p''_{I,J-1} + b_{I,J+1}p''_{I,J+1} + b''_{I,J} \quad (3.15)$$

Anche l'algoritmo PISO può essere modificato per accogliere il termine relativo alla soluzione di un caso instazionario. In questo caso, prendere in considerazione i termini temporali, converge nell'aggiunta di elementi propri del passo temporale precedente. In particolare, durante la discretizzazione le equazioni della conservazione della quantità di moto risultano:

$$\begin{aligned} b_{i,j}u_{i,j}^{***} &= \sum b_{n,b}u_{n,b}^{**} + \rho_P^0 \frac{\Delta V}{\Delta t} + (p_{I-1,j}^{***} - p_{I,j}^{***})T_{i,j} + (c_{i,j} + b_P^0 u_P^0) \\ b_{I,j}v_{I,j}^{***} &= \sum b_{n,b}v_{n,b}^{**} + \rho_P^0 \frac{\Delta V}{\Delta t} + (p_{I,j-1}^{***} - p_{I,j}^{***})T_{i,j} + (c_{I,j} + b_P^0 v_P^0) \end{aligned} \quad (3.16)$$

Per quanto riguarda l'equazione della pressione, la modifica consiste nell'aggiunta del termine  $(\rho_P^0 - \rho_P) \frac{\Delta V}{\Delta t}$  al termine sorgente presente nell'equazione, ottenendo:

$$b_{I,J}p''_{I,J} = b_{I+1,J}p''_{I+1,J} + b_{I-1,J}p''_{I-1,J} + b_{I,J-1}p''_{I,J-1} + b_{I,J+1}p''_{I,J+1} + b''_{I,J} + (\rho_P^0 - \rho_P) \frac{\Delta V}{\Delta t} \quad (3.17)$$

Schematicamente, gli algoritmi PISO e PISO unsteady sono riportati nel diagramma sottostante:

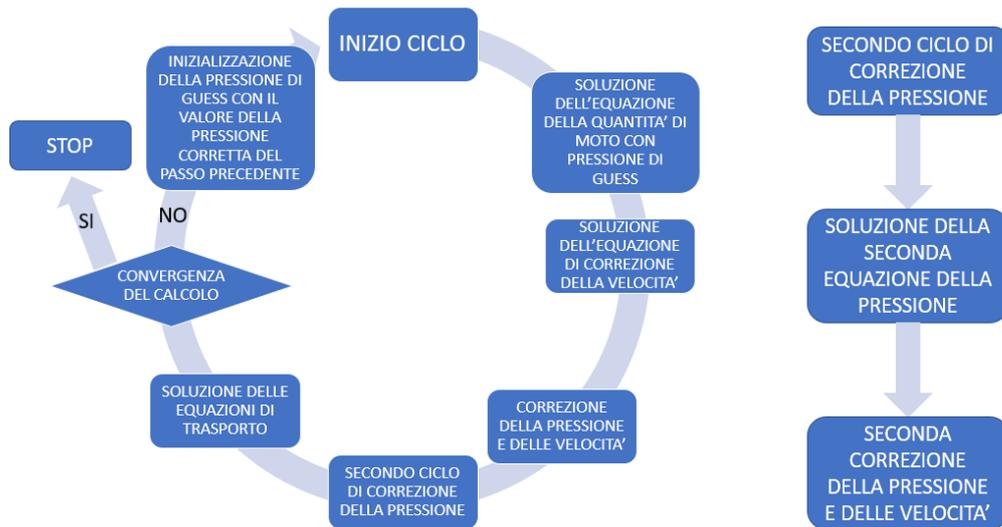


Figure 3.3: Schema algoritmo PISO [9]

### 3.3.4 L'algoritmo PIMPLE

La combinazione degli algoritmi SIMPLE e PISO confluisce in un terzo algoritmo ampiamente utilizzato, l'algoritmo PIMPLE [13]. Lo sviluppo di questo schema è dato dalla necessità di ottimizzare i due algoritmi per i casi di flusso instazionario che garantisca una migliore precisione nel calcolo del campo di pressione corretto.

L'idea di base di questa metodologia è la soluzione, per ogni time-step, delle equazioni tramite la procedura SIMPLE all'interno del quale più cicli di correzione della pressione vengono svolti, in accordo con l'algoritmo PISO. I termini fondanti sono gli *OuterCorrectors* e gli *InnerCorrectors*, che sono due termini che vengono settati dall'utente.

I primi rappresentano il numero di volte in cui tutto il sistema di equazioni viene risolto, partendo dal *guess* delle velocità fino alla correzione delle pressioni, in accordo con l'algoritmo SIMPLE. In ognuno dei cicli esterni, perciò, viene identificata una pressione di *guess*, viene risolta l'equazione della quantità di moto e vengono trovati i termini correttivi. La pressione di *guess* dei cicli successivi al primo è la pressione calcolata alla fine del ciclo correttivo ultimo.

Gli *Inner Correctors*, invece, rappresentano il numero di cicli di correzione della pressione che devono venire svolti all'interno di un *OuterCorrector*, in accordo con l'algoritmo PISO. Il ciclo *OuterCorrectors* e *Inner Correctors* viene richiamato per ogni time step dopo che tutte la soluzione di tutte le equazioni ha raggiunto la convergenza richiesta. Lo schema logico di funzionamento dell'algoritmo *PIMPLE* è contenuto in figura 3.4.



Figure 3.4: Schema algoritmo PIMPLE

### 3.3.5 Il caso bifase e l'approccio VOF e l'approccio Eulerian multiphase

Un flusso bifase è un flusso nel quale sono contenuti due o più materiali la cui fase è diversa. Nella maggior parte dei casi sono situazioni in cui una miscela aeriforme interagisce con una miscela liquida ma è anche possibile avere la necessità di modellare il comportamento del particolato solido in un gas o in un liquido. Nel caso specifico di simulazioni in ambito navale vengono simulate sia la fase acquosa, sia la fase aeriforme.

Per la modellazione matematica di una condizione bifase si ricorre ad una funzione che controlla la percentuale in volume delle due fasi in ogni cella. L'obiettivo è quindi quello di calcolare il grado di "riempimento" della cella; se è riempita al 100% dalla fase aeriforme, se invece, è riempita al 100% dalla fase liquida oppure se è riempita in parte da una ed in parte dall'altra, come, per esempio, le celle all'interfaccia tra acqua ed aria nelle simulazioni in ambito navale.

L'equazione che si aggiunge al sistema di equazioni di base, è la seguente:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\vec{U} \gamma) = 0 \quad (3.18)$$

Dove:

- $\vec{U}$  è il campo di velocità dei due fluidi;
- $\gamma$  è la frazione volumica, anche chiamata in letteratura *phase fraction*;  $\gamma$  può assumere valori compresi tra 0 ed 1. Quando la frazione volumica assume i valori estremi 0 ed 1 allora la cella di interesse si trova in una zona con un'unica fase.

Per le celle che contengono entrambe le fasi, e quindi per le celle in cui  $\gamma$  acquisisce un valore  $0 < \gamma < 1$  le caratteristiche del fluido sono calcolate come media pesata sulla frazione volumica delle due fasi in accordo con le due relazioni di seguito elencate:

$$\begin{aligned} \rho &= \rho_l \gamma + \rho_g (1 - \gamma) \\ \mu &= \mu_l \gamma + \mu_g (1 - \gamma) \end{aligned} \quad (3.19)$$

La difficoltà nell'utilizzo di questo modello fa riferimento a garantire la conservazione della massa delle due fasi durante la simulazione. Infatti, a causa dell'introduzione della funzione di *volume fraction*, vengono introdotte nuove sorgenti di errori causate dall'approssimazione numerica delle equazioni.

Per risolvere le problematiche descritte, in OpenFoam viene implementata una modifica all'equazione iniziale, per far fronte alle problematiche numeriche introdotte. Il modello che si ottiene viene definito: modello Euleroiano per un flusso bifase. In questo modello, le equazioni della *phase fraction* di ogni fase, gassosa  $g$  e liquida  $l$  sono risolte separatamente e sono descritte dalle seguenti relazioni:

$$\begin{aligned} \frac{\partial \gamma}{\partial t} + \nabla \cdot (U_l \gamma) &= 0 \\ \frac{\partial(1 - \gamma)}{\partial t} + \nabla \cdot [(U_g(1 - \gamma))] &= 0 \end{aligned} \quad (3.20)$$

Assumendo che la velocità globale del fluido sia data dalla media ponderata delle velocità del gas e del liquido,  $\vec{U} = \gamma \vec{U}_l + (1 - \gamma) \vec{U}_g$  l'equazione per l'evoluzione della frazione in volume, 3.20.a, diventa:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (U \gamma) + \nabla \cdot [U_r \gamma (1 - \gamma)] = 0 \quad (3.21)$$

Dove il termine  $U_r = U_l - U_g$  è la *compression velocity* e rappresenta il vettore delle velocità relative tra le due fasi.

Confrontando l'equazione 3.21 con l'equazione di partenza per lo studio del Volume of Fluid, l'equazione 3.18, compare il termine  $\nabla \cdot [U_r \gamma (1 - \gamma)]$  che rappresenta un termine convettivo per la frazione in volume che ha come scopo e come effetto la compressione della *free surface* nel modello multifase. L'aggiunta al modello di questo termine assicura una migliore definizione della superficie di contatto tra le due fasi e una migliore conservazione della frazione in volume delle fasi presenti nel modello.

Anche l'equazione della quantità di moto subisce una modifica, in particolare, si aggiungono gli effetti della tensione superficiale tra i due fluidi. Questa entità genera un ulteriore gradiente di pressione che genera una forza, definita, costante, come:

$$f_\sigma = \sigma \kappa \nabla \gamma \quad (3.22)$$

dove  $\kappa$  è la curvatura locale della superficie di contatto all'interfaccia ed è definita come:

$$\kappa = -\nabla \cdot \left( \frac{\nabla \gamma}{|\nabla \gamma|} \right) \quad (3.23)$$

Osservando un sistema multifase, è importante considerare che è presente il gradiente idrostatico, dato dal peso del fluido. Per semplificare l'equazione

della pressione, quindi, si ragiona in termini di pressione specifica,  $p_s$ , definita come:

$$p_s = p - \rho \vec{g} \cdot x \quad (3.24)$$

Introducendo la variabile nell'equazione della quantità di moto si ottiene:

$$\frac{\partial(\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \vec{U}) - \nabla \cdot (\nu \nabla \vec{U}) - (\nabla \vec{U}) \cdot \nabla \gamma = -\nabla p_s - \vec{g} \cdot \vec{x} \nabla \rho + \sigma \kappa \nabla \gamma \quad (3.25)$$

Il sistema di equazioni di Navier-Stokes, adattate al caso multifase, incomprimibile, è composto dalle seguenti equazioni:

$$\begin{aligned} \nabla \cdot U &= 0 \\ \frac{\partial \gamma}{\partial t} + \nabla \cdot (U \gamma) + \nabla \cdot [U_r \gamma (1 - \gamma)] &= 0 \\ \frac{\partial(\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \vec{U}) - \nabla \cdot (\nu \nabla \vec{U}) - (\nabla \vec{U}) \cdot \nabla \gamma &= -\nabla p_d - \vec{g} \cdot \vec{x} \nabla \rho + \sigma \kappa \nabla \gamma \end{aligned} \quad (3.26)$$

Nelle situazioni di flusso multifase, dalla soluzione numerica delle equazioni per le due fasi, nasce un'ulteriore problematica identificata come *ventilazione numerica*. La ventilazione numerica è un fenomeno che si manifesta durante le simulazioni che implementano gli algoritmi VOF.

La ventilazione numerica nasce dall'approssimazione non precisa della percentuale in volume della fase preponderante. Prendendo in esempio gli oggetti galleggianti sull'acqua, la ventilazione numerica si manifesta quando alcune aree della superficie del corpo che dovrebbero essere "bagnate" da un fluido al 100%, sono affacciate ad un fluido la cui densità è minore. La conseguenza è che le forze agenti su un corpo affetto da ventilazione numerica risultano minori, per via della densità, rispetto alle forze reali o con ventilazione assente. È dimostrato che questo problema si risolve svolgendo una discretizzazione spaziale piuttosto raffinata nelle zone critiche del flusso e utilizzando un time-step ridotto.

### 3.4 La turbolenza

Durante gli studi fluidodinamici, per avere una visione completa delle fenomenologie che intervengono, si approfondiscono e si implementano delle considerazioni sulla turbolenza. La turbolenza è un fenomeno che, nell'ambito della fluidodinamica, causa una variazione spaziale e temporale delle grandezze in

gioco, velocità e pressione, di tipo semi-randomico.

La turbolenza viene implementata nelle equazioni di Navier-Stockes di base con la cosiddetta decomposizione alla Reynolds. Secondo questo approccio le grandezze del fluido, velocità e pressioni, vengono scomposte in due termini, un termine che rappresenta il valore medio della grandezza stessa ed uno che tiene conto delle fluttuazioni del termine in esame. Le grandezze scomposte sono:

$$\begin{aligned} p &= \bar{p} + p'; \\ u &= \bar{u} + u'; \\ v &= \bar{v} + v'; \\ w &= \bar{w} + w'; \end{aligned} \tag{3.27}$$

Sostituendo il concetto della decomposizione alla Reynolds all'interno delle equazioni di Navier-Stockes si ottengono le Reynolds Averaged Navier Stockes (RANS) equations che, se risolte, permettono di calcolare i valori medi e fluttuazioni dei campi di velocità e di pressione, permettendo di risolvere in maniera completa la turbolenza, in termini di valori medi e fluttuazioni, presente nel sistema. Svolgendo i passaggi per l'equazione della quantità di moto e seguendo una notazione in cui i pedici identificano la componente vettoriale, si ottiene la seguente equazione:

$$U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j^2} - \frac{\partial(u_i \bar{u}_j)}{\partial x_j} \tag{3.28}$$

Dove il termine  $U_x$  è il termine di velocità media ed il termine  $u_x$  rappresenta la corrispondente fluttuazione.

Osservando la 3.28 si nota la differenza con il caso di flusso laminare. Il termine che si aggiunge, dopo la mediazione delle equazione, è il termine che rappresenta il flusso di quantità di moto dovuto alle fluttuazioni turbolente di velocità. Questo termine si può riformulare considerando che  $u_i \bar{u}_j$  ha la grandezza di uno stress, a tal proposito si introduce il termine di *eddy viscosity*,  $\nu_T$ :

$$u_i \bar{u}_j = \nu_T \frac{\partial \bar{U}_i}{\partial x_j} \tag{3.29}$$

La soluzione delle equazioni mediate alla Reynolds risulta computazionalmente più costosa, a causa dell'introduzione di ulteriori incognite, in particolare per le componenti fluttuanti delle grandezze in gioco.

Le tecniche per integrare nel codice di base la turbolenza durante lo studio tramite la fluidodinamica computazionale sono diverse e differiscono per la complessità e per il costo computazionale. La tecnica più complessa che prevede la soluzione completa della turbolenza partendo dalla soluzione delle equazioni di Navier-Stokes di base è la DNS, *Direct Navier-Stokes*; come si può intendere dal nome questo metodo risolve il flusso in maniera completa considerando, in funzione della discretizzazione spaziale e temporale utilizzata, tutte le scale di turbolenza.

In alternativa, per rendere il calcolo più veloce, sono stati studiati dei modelli che riescono ad introdurre gli effetti della turbolenza senza ricorrere alla soluzione diretta delle equazioni. Questi calcolano solo alcune grandezze caratteristiche della turbolenza e non tutte le fluttuazioni di velocità e pressione. I modelli chiamati *eddy viscosity models* sono ampiamente utilizzati nei codici di calcolo CFD per la loro efficacia e il loro costo limitato. Altri metodi che non risolvono in maniera completa la turbolenza ma solo alcune caratteristiche principali proprie del campo di moto in esame, attualmente molto utilizzati in campo ingegneristico sono: il modello  $k-\epsilon$  ed il modello  $k-\omega$ . Nel seguito viene descritto nel dettaglio il primo dei due elencati dato il suo utilizzo nelle simulazioni svolte.

### Modello $k-\epsilon$

Il modello di calcolo della turbolenza  $k-\epsilon$  è un modello ampiamente utilizzato in ambito ingegneristico date le sue buone capacità di calcolo e il suo costo computazionale relativamente basso rispetto a modelli come il modello *LES* e *DNS*. Il  $k-\epsilon$  permette di tenere in considerazione gli effetti della convezione e diffusione, e produzione e distruzione della quantità di moto dovuta alle fluttuazioni turbolente risolvendo due equazioni di trasporto, rispettivamente una per l'energia cinetica turbolenta  $k$  ed una per il rateo di dissipazione dell'energia turbolenta  $\epsilon$ . Le due equazioni implementate nel modello sono le seguenti:

$$\begin{aligned} \frac{(\rho\kappa)}{\partial t} + \nabla(\rho\kappa\vec{U}) &= \nabla \left[ \frac{\nu_t}{\sigma_k} \nabla k \right] + 2\nu_t S_{ij} \cdot S_{ij} - \rho\epsilon \\ \frac{(\rho\epsilon)}{\partial t} + \nabla(\rho\epsilon\vec{U}) &= \nabla \left[ \frac{\nu_t}{\sigma_\epsilon} \nabla \epsilon \right] + 2\nu_t C_{1\epsilon} \frac{\epsilon}{\kappa} S_{ij} \cdot S_{ij} - C_{2\epsilon} \rho \frac{\epsilon^2}{\kappa} \end{aligned} \quad (3.30)$$

Dove i termini  $C_\nu$ ,  $\sigma_\kappa$ ,  $\sigma_\epsilon$ ,  $C_{1\epsilon}$  e  $C_{2\epsilon}$  sono costanti che vengono inserite in input nel codice di calcolo e garantiscono un controllo per le diverse tipologie di turbolenza. Inoltre,  $S_{ij}$  è il tensore del rateo di deformazione della particella in esame.

Nelle equazioni compare la *eddy viscosity*  $\nu_t$ , che si può calcolare svolgendo un'analisi dimensionale sulle scale della turbolenza stessa, scale di velocità  $\theta$  e le scale spaziali  $l$ . Queste ultime si possono relazionare all'energia cinetica turbolenta e al rateo di dissipazione come segue:

$$\theta = \kappa^{1/2} \quad l = \frac{k^{3/2}}{\epsilon} \quad (3.31)$$

e la *eddy viscosity*:

$$\nu_t = C\rho\theta l = \rho C_\nu \frac{k^2}{\epsilon} \quad (3.32)$$

Infine, attraverso la relazione di Boussinesq, si ottengono gli stress di Reynolds contenuti nelle equazioni di Navier-Stockes mediate alla Reynolds. La relazione applicata in ultimo passaggio è la seguente:

$$-u_i \bar{u}_j = \nu_t \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{2}{3} \rho k \delta_{ij} = 2\nu_t S_{ij} - \frac{2}{3} \rho k \delta_{ij} \quad (3.33)$$

Dove  $\delta_{ij}$  è la funzione *Delta di Dirach*, che vale 1 quando  $i = j$  mentre vale 0 quando  $i \neq j$ . Il calcolo di  $u_i u_j$  permette di completare l'equazione mediata della quantità di moto che è quindi possibile risolvere tenendo in considerazione anche il flusso di quantità di moto dovuto alle fluttuazioni turbolente.

### 3.5 Movimento della mesh

Il movimento dei corpi sotto l'azione delle forze fluidodinamiche applicate richiede la movimentazione dei punti della mesh in accordo con il moto del corpo in esame. La movimentazione dei punti della griglia di calcolo in accordo con il movimento del corpo, è un passaggio che può essere ottenuto mediante diverse logiche.

In letteratura, due sono i metodi attuabili:

- Il metodo *Morphing Mesh*, secondo il quale, in funzione del movimento del corpo si ha una movimentazione dei punti della mesh e quindi si ha una deformazione delle celle circostanti il corpo;

- Il metodo *Overset Mesh*. Per l'implementazione di questo sistema si prevedono due ambienti di calcolo separati, uno è quello che di solito, per una "regola non scritta", viene chiamato *tank* o *background* e rappresenta i limiti del dominio ed uno è un blocco che circonda il corpo e si muove in maniera solidale con il corpo stesso; in questo caso la mesh non viene deformata, rispetto alla condizione iniziale, dal movimento del corpo ma è tutta la mesh di *overset* che si muove mantenendo intatta la discretizzazione al suo interno. Il punto importante su cui si costruisce questo approccio è lo scambio di dati tra il *tank* e l'*overset*; sul bordo di quest'ultimo, infatti avvengono i processi di interpolazione dei dati delle celle tra i due ambienti ed è una fase molto importante per la convergenza della simulazione.

### 3.5.1 L'approccio Overset Mesh

Dato l'utilizzo approfondito della logica di *oversetMesh* per le simulazioni del progetto in esame, si approfondiscono i concetti relativi all'implementazione ed al funzionamento della mesh *overset* in OpenFoam. [14]

Il principio di base su cui si fonda questa metodologia è la soluzioni delle equazioni nel dominio di *background* e nel dominio di *overset*. La complicazione risiede nel fatto che le celle di *background* che si sovrappongono all'*overset* vengono rimosse dal dominio di calcolo e, al loro posto, si interpolano i valori delle grandezze fisiche al bordo dell'*overset*. Ogni cella contenuta nella zona tra contorno dell'*overset*, sovrapposizione tra *overset* e *background* e zona in cui è contenuto il corpo assume un valore ed una caratteristica precisa tra le tre possibilità di seguito elencate:

**Calculated** Assumono valore 1; in queste celle le equazioni vengono risolte normalmente;

**Interpolated** Assumono valore 2; sono le celle dove avviene l'interpolazione. La zona di sovrapposizione tra le celle di *overset* e le celle del *tank* è una zona in cui le celle è indispensabile che abbiano dimensioni simili per agevolare l'interpolazione dei dati. A loro volta queste celle possono assumere due funzioni: le celle *Donors* sono le celle da cui i dati provengono mentre le celle *Acceptors* sono le celle che acquisiscono il dato per trasferirlo poi alle celle di tipo *Calculated*.

**Holes** Assumono valore 0; sono le celle in cui la soluzione non viene calcolata.

L'assegnazione del tipo di cella e l'interpolazione dei dati è il passaggio che contraddistingue il codice, questa operazione si può ottenere in diversi modi descritti in letteratura con i seguenti nomi:

- cellVolumeWeight;
- inverseDistance;
- leastSquares;
- trackingInverseDistance;

L'implementazione in Openfoam utilizzata prevede l'utilizzo del metodo **inverse distance** e di seguito verrà approfondito.

### L'interpolazione di tipo *InverseDistance*

Con questo metodo le celle *Interpolated* sono le celle *Acceptor*, mentre le celle *donors* sono celle di tipo *calculated* che vengono definite a partire dalle celle *acceptor*. Le celle donatrici vengono trovate dall'algoritmo confrontando gli indici delle celle di *overset* e *background* e la vicinanza dei punti delle celle corrispondenti, le celle tra di loro più vicine vengono dichiarate come *accettore* e *donatore*. Definite le funzioni delle celle vengono calcolati i pesi di ogni donatore, e durante questo passaggio entra in gioco il concetto di *inverse distance*. Infatti il peso del singolo donatore viene definita come:

$$p_i = \frac{1}{\frac{|d_i|}{P}} \quad (3.34)$$

Dove il termine  $|d_i|$  è il modulo del vettore congiungente il centro cella del donatore con il centro cella dell'accettore ed il termine  $P$  è dato dalla sommatoria delle distanza *inverse*:  $P = \sum_n^i \frac{1}{|d_i|}$ .

La grandezza del fluido che viene "passata" tra le due mesh viene quindi ponderata tramite la funzione peso definita dal metodo *inverseDistance*.

In funzione dell'algoritmo le celle donatrici per ogni cella accettore possono essere più o meno.

## 3.6 Struttura macroscopica di un codice CFD

Un codice CFD è quindi l'insieme di equazioni, condizioni al contorno, e algoritmi risolutivi utili a risolvere le equazioni stesse per una situazione particolare che rappresenta il caso studio in esame.

Le fasi principali della costruzione di un codice per l'analisi tramite la fluidodinamica computazionale sono:

- **Pre-Processing**

1. **Definizione del dominio di calcolo.** Durante questa operazione si costruiscono le regioni caratteristiche dell'ambiente del quale si vuole studiare la fluidodinamica, insieme ad esse è importante definire i sistemi di riferimento in funzione delle esigenze dello studio.
2. **Generazione della griglia di calcolo.** Anche chiamata *mesh* nella letteratura anglosassone, è la suddivisione del dominio di calcolo in parti più piccole necessarie a definire in maniera discreta i luoghi fisici nei quali risolvere le caratteristiche del fluido tramite la soluzione numerica delle equazioni del modello. La generazione della mesh è un passaggio fondamentale che può compromettere la corretta convergenza del codice.

L'introduzione della discretizzazione del sistema provoca, inevitabilmente un certo errore ma è indispensabile per la risoluzione delle equazioni. Nell'ottica di minimizzare l'errore, quindi, è importante capire le zone critiche del sistema in esame e dettagliare, in determinati punti, la suddivisione attraverso dei raffinamenti delle celle.

3. **Settaggio delle condizioni al contorno.** Nell'assegnazione delle condizioni al contorno, ad ogni superficie del dominio di calcolo si assegna un valore caratteristico delle grandezze incognite o delle derivate delle grandezze incognite. In funzione del software che si utilizza sono disponibili svariate tipologie di condizioni al contorno.
4. **Definizione delle proprietà del fluido o dei fluidi** presenti nel dominio di calcolo di cui si vogliono studiare le caratteristiche.
5. **Impostazione dei modelli fisici** relativi ai fenomeni presenti nel dominio di calcolo; possono essere presenti situazioni di flusso

multifase, comprimibile o incomprimibile, o nel quale sono presenti reazioni chimiche o particolari condizioni di turbolenza che si vogliono trovare. In questa fase si definiscono i modelli per il calcolo delle specifiche condizioni in cui si simula il caso studio.

6. **Settaggio dei solutori.** Ogni algoritmo risolutivo richiede una serie di dati di input che guidano il codice nella soluzione delle equazioni. I settaggi si riferiscono agli schemi di calcolo da attuare per la soluzione delle equazioni, alle tolleranze nella soluzione e ai cosiddetti *stopping criteria* ossia dei punti in cui si reputa raggiunta la convergenza.

- **Processing**

Nella fase di processing il calcolo viene eseguito, si calcolano le iterazioni previste dai settaggi relativi agli schemi risolutivi fino a che il codice raggiunge i criteri di fine calcolo.

- **Post processing**

Durante il post processing si attuano le procedure per ottenere le grandezze derivate dai valori di velocità e pressioni calcolati; nella maggior parte dei casi è necessario ottenere le forze applicate al corpo oppure il flusso termico, altri dati che si possono ottenere tramite algoritmi dedicati durante il calcolo sono relativi al movimento dei corpi soggetti alle forze aerodinamiche.

Durante il post-processing oltre ai dati inerenti al sistema fisico è necessario osservare i dati relativi alla numerica della simulazione per capire le modalità di convergenze ottenute durante il calcolo; si osservano i residui, il numero di Courant per le simulazioni con evoluzione temporale, i valori della funzione  $y^+$  per le simulazioni con flusso di parete ed altre grandezze che possono variare in funzione del caso studio.

## 3.7 Convergenza ed analisi della simulazione

La convergenza della simulazione CFD è la condizione in cui il calcolo restituisce il risultato che rappresenta il risultato finale e definitivo che non ha più un'ampia variabilità rispetto alle iterazioni.

La convergenza della simulazione CFD viene valutata attraverso l'analisi di alcuni parametri numerici fondamentali del codice di calcolo e dei risultati fisici in uscita dal calcolo.

## Residui

Il controllo della convergenza tramite i residui, avviene già durante il calcolo. Le iterazioni interne per la velocità e per la pressione sono controllate e si fermano al raggiungimento della tolleranza scelta nel foglio di calcolo *fvSolution*. È presente un'oscillazione del valore dei residui finali durante il calcolo che non rappresenta un limite alla convergenza in quanto comunque il valore rispetta il limite di tolleranza imposto in input sempre nella dictionary *fvSolution*.

## Grandezze fisiche

Il secondo indice di convergenza della simulazione è rappresentato dalle grandezze fisiche calcolate dalla simulazione. In questo caso, le grandezze fisiche da tenere sotto controllo sono le forze applicate alla carena e gli spostamenti della carena.

La simulazione, nonostante l'impostazione di soluzione temporale instazionaria, ha l'obiettivo di calcolare uno stato di equilibrio finale stazionario, perciò, sia le forze che le grandezze relative al moto della barca, a convergenza della simulazione raggiunta, diventano anch'esse costanti nel tempo, a meno di piccole oscillazioni. Per la dinamica della barca, un ulteriore controllo sullo stato di equilibrio, giunge dall'analisi delle accelerazioni e delle velocità, che, raggiunto il punto di stabilità devono essere entrambe azzerate.

### 3.7.1 Validazione del modello

In seguito alla convergenza delle simulazioni, un aspetto molto importante durante uno studio fluidodinamico, è la comprensione della corretta convergenza. Infatti, la simulazione potrebbe raggiungere la convergenza a risultati scorretti se la discretizzazione del dominio ed i settaggi non sono corretti.

Per far fronte a questa problematica, quando si studia un nuovo modello, è indispensabile confrontare i dati della simulazione con dei dati ottenuti, per le stesse identiche condizioni di input, da altre tipologie di studi. Per le analisi di tipo navale un'alternativa all'CFD è rappresentata dai VPP, meno costosi ma che danno risultati più grossolani, oppure dalle prove sperimentali in vasca che, invece, sono molto costose ma ricostruiscono la situazione di navigazione reale e quindi rappresentano una buona alternativa alla fluidodinamica computazionale. Nonostante ciò, però, le prove in vasca rappresentano spesso un costo eccessivo da sostenere all'interno di un progetto e quindi questa tipologia di studi non viene presa in considerazione in ambito commerciale.



## Chapter 4

# Open Foam, Paraview e la struttura della simulazione

*Svolgere gli studi fluidodinamici tramite la Computational Fluid Dynamics, è un procedimento complesso ed articolato.*

*Per applicare nel concreto la fluidodinamica computazionale, si utilizzano dei software che sono costruiti in maniera tale da permettere all'utente di gestire solo una piccola parte, quella che può differire maggiormente da caso a caso, delle variabili in gioco in un codice completo: condizioni al contorno, geometrie, tipologie di solutori sono a discrezione dell'utente, la loro implementazione e la successiva applicazione avviene in maniera automatica all'interno di un algoritmo.*

*Numerosi sono i software disponibili; quelli definiti commerciali sono disponibili tramite l'acquisto delle licenze, sono sviluppati in maniera completa e sono più semplici da utilizzare, nonché corredati da una ricca documentazione. I software open source, invece, sono codici sviluppati da diversi utenti e resi disponibili a tutti senza la necessità delle chiavi di licenza per poterli utilizzare. Il loro utilizzo è più complesso ma numerosi studi hanno già dimostrato una validità soddisfacente nel complesso, data anche la possibilità di modificare l'implementazione del codice sorgente e di creare modelli personalizzati.*

*Il seguente capitolo è dedicato alla descrizione di OpenFoam, il software open source più importante disponibile attualmente, verranno descritte le logiche alla base del software, le sue caratteristiche principali di interfaccia con l'utente e l'implementazione del codice utilizzato per la soluzione del caso incomprimibile e multifase. I contenuti del capitolo sono tratti da [8] [7] [5].*

## 4.1 OpenFoam, il software opensource per la Computational Fluid Dynamics

OpenFoam è un codice per la Computational Fluid Dynamics di tipo open source che implementa diversi modelli di calcolo per la soluzione delle equazioni di Navier-Stokes adatte per varie tipologie di flusso fluido.



Figure 4.1: OpenFoam logo [8]

OpenFoam è un codice scritto in C++ con orientamento ad oggetti ed è strutturato come segue: i files sono raggruppati all'interno di sottocartelle che contengono ognuna ulteriori gruppi di cartelle suddivise per argomento o funzione. Si trovano:

- La cartella */applications* contenente i codici "finali" che, una volta lanciati svolgono una determinata operazione;
- La cartella */src* nella quale sono raggruppate tutte le librerie indispensabili ai codici dei solvers o delle utilities, descritte nel seguito;
- La cartella */Tutorials* nella quale si possono trovare una serie di simulazioni svolte utilizzando tutti i diversi solutori, che aiutano e guidano l'utente nel settaggio della simulazione per il determinato caso studio.

Nelle altre cartelle, meno importanti per lo svolgimento effettivo della Computational Fluid Dynamics attraverso OpenFoam si possono trovare documentazioni, contributi da terze parti ed altri documenti di riepilogo.

Osservando il codice con un approccio gerarchico, in cima alla scala si trovano i *solvers* all'interno dei quali è presente il *main code*. I passaggi all'interno del main code richiamano delle funzioni contenute nelle librerie, rappresentate dagli *Header Files*, con estensione *.H*, che sono delle entità proprie del linguaggio C++ che contengono delle funzioni, insiemi di operazioni che vengono svolte per un determinato obiettivo. Ogni funzione, all'interno del proprio algoritmo, può a sua volta invocare altre funzioni della stessa libreria o di altre librerie contenute in diverse cartelle del software. Parallelamente ai *Solvers* che sono gli algoritmi per la fase di Processing, ci sono le *Utilities*, algoritmi necessari alle altre fasi di una simulazione fluidodinamica.

L'elemento fondamentale del software, che interfaccia l'utente al codice, è rappresentato dalle *dictionaries* che sono le entità attraverso le quali si svolgono i settaggi del codice; ognuna di queste si riferisce ad una determinata operazione che richiede degli input in ingresso. Nel seguito vengono descritte più dettagliatamente le entità fondamentali del software.

### 4.1.1 Utilities

Le *Utilities* sono raggruppate nella cartella *applications/Utilities/* raggruppano tutti gli algoritmi contenenti i passaggi che servono per le fasi di *pre-processing* e di *post-processing*. Nella cartella descritta raggruppate in sottocartelle ordinate per macro-funzione. Per esempio, per discretizzare il dominio si utilizzano le utilities *blockMesh* oppure *sappyHexMesh* contenute nella cartella *applications/utilities/mesh/generation*.

### 4.1.2 I solutori

Un codice di calcolo CFD, solitamente, è strutturato per studiare più situazioni anche molto diverse tra loro; questo concetto è comunemente indicato con il termine *multi purpose*. In termini pratici si può spiegare il concetto pensando che con lo stesso software si possono studiare profili alari che si muovono in aria, oggetti immersi contemporaneamente in due fluidi, miscele gassose eterogenee, fenomeni termodinamici ecc...

Per poter fare tutto ciò in Openfoam sono disponibili diversi *solutori*. I solutori quindi sono degli algoritmi che, in funzione del caso studio, sono ottimizzati per effettuare il calcolo in determinate condizioni specifiche del caso studio definito dall'utente.

Un solutore è contenuto in un file di estensione *.C* che, al suo interno contiene una serie di passaggi che per lo più richiamano una serie di funzioni contenute nelle librerie. Questo approccio "modulare" serve a raggiungere una programmazione più ordinata e accessibile, in maniera tale da facilitarne la comprensione e la modifica.

I solutori in OpenFoam più utilizzati sono i solutori *simpleFoam*, *pimpleFoam*, *icoFoam* che risolvono un caso monofase incomprimibile stazionario o instazionario e con la presenza o meno della turbolenza. Mentre i solutori *interFoam* ed *overInterDyMFoam* sono dedicati al calcolo in presenza di due fluidi incompressibili, immiscibili e isotermi tramite l'utilizzo del metodo *Volume Of Fluid* per la soluzione dell'interfaccia tra i due fluidi. La differenza tra i due consiste nella possibilità di implementazione di diversi modelli di

movimento della mesh.

Tanti altri sono i *solvers* disponibili in OpenFoam, per simulare fenomeni termici, di flussi compressibili oppure per fenomeni di combustione e la caratteristica principale su cui si basa questo codice open source risiede nella possibilità di modificare il codice sorgente dei solvers per adattarlo ulteriormente al caso studio specifico.

Nel seguito viene approfondito il solutore utilizzato per la realizzazione degli studi effettuati sull'imbarcazione oggetto della presente tesi.

### 4.1.3 Le librerie

Le librerie, generalmente nel linguaggio di programmazione in esame, il *C++*, sono entità rappresentate da un documento di estensione *.H* che vengono utilizzate per contenere una serie di funzioni. Una funzione è un "sotto-algoritmo" di passaggi di base che serve all'esecuzione di una determinata operazione. Nelle funzioni sono contenuti tutti i passaggi elementari per l'esecuzione dell'operazione. Le librerie sono contenute nella cartella */src* nella quale le librerie sono raggruppate per funzioni in numerose sottocartelle. Per esempio, le funzioni contenenti i passaggi per i calcoli inerenti alla dinamica del corpo sono contenuti nella cartella *src/dynamicFoam* e le librerie vengono richiamate nelle dictionary o all'interno dei codici dei solvers o delle utilities.

### 4.1.4 Le Dictionaries

Le *dictionaries*, utilizzando il linguaggio proprio del software, sono gli strumenti che l'utente può manipolare al fine di settare la simulazione. Le diverse *dictionaries* sono organizzate in maniera tale che ognuna raccolga la lista di *keywords* e relativi parametri, specifici per una specifica categoria di dati in input. Per esempio, è presente la dictionary per tipologie di mesh, *blockMeshDict* o *snappyHexMeshDict* la dictionary per gli schemi risolutivi delle equazioni e così via.

Tramite le *Dictionaries* si controllano le *Utilities* e le *Solvers*.

## 4.2 La simulazione CFD implementata su Open-Foam

La struttura della simulazione, rispetto all'utente, è formata da più cartelle e sottocartelle contenenti i fogli di testo nei quali sono indicate le voci per il settaggio delle varie parti del codice sorgente e i valori di input dell'algoritmo utilizzato.

Le tre cartelle sempre e necessariamente presenti all'interno della simulazione sono:

- Cartella *constant*, per la raccolta dei file per la mesh, geometrie utilizzate e altri oggetti come i fogli per il controllo della turbolenza, delle caratteristiche dei fluidi utilizzati e la dictionary per la dinamica del corpo;
- Cartella *system*, contiene le cosiddette *dictionaries* relative al controllo delle applicazioni ed ai solutori utilizzati nella simulazione. Per esempio: *blockMeshDict* costruisce un parallelepipedo già discretizzato, *controlDict* è, come suggerisce il nome, l'oggetto che controlla il codice di calcolo vero e proprio, *fvSolution* e *fvSchemes* controllano la numerica del codice. In funzione delle necessità, possono essere aggiunte *dictionaries* relative alla mesh ed al post-processing.
- Cartella *0* o *time directories*, per le condizioni al contorno nell'istante iniziale (in questo caso 0). Durante il funzionamento del solutore si può decidere di salvare la soluzione in determinate iterazioni o istanti temporali. Il salvataggio crea una cartella rinominata con il valore dell'iterazione o del passo temporale.

Altre due cartelle che vengono create durante il funzionamento dell'algoritmo risolutivo e sono presenti nella cartella principale della simulazione sono:

- Cartella *postProcessing*, in questa cartella sono raggruppati i report che si ottengono dalle applicazioni del codice dedicate al post processing; i dati si riferiscono alle forze applicate ai corpi in esame, ai coefficienti di forza oppure possono indicare i valori inerenti ai parametri numerici della simulazione come Courant e  $y^+$ .
- Cartelle *Processor n*, nei casi in cui si parallelizza il calcolo, vengono create il numero di cartelle pari ai processori utilizzati. Queste cartelle contengono, a loro volta, la cartella *constant* per l'*n-esimo* processore

contenente il blocco di mesh che il suddetto processore deve risolvere, e le cartelle temporali.

La gestione del codice avviene tramite scrittura dei fogli di testo contenuti nelle cartelle elencate sopra e invocazione del codice principale tramite linea di comando.

### 4.2.1 Il codice `InterFoam` e `overInterDyMFoam`

I solutori *interFoam* e *overInterDyMFoam* sono algoritmi implementati in OpenFoam per la soluzione di flussi incomprimibili, bifase e in cui le due fasi sono immiscibili. Il primo dei due è un modello adatto anche all'introduzione dei codici per la dinamica di eventuali corpi immersi nel fluido in esame che funzionano con la logica di morphing mesh. Il secondo dei due è ottimizzato all'introduzione della dinamica attraverso l'approccio di overset mesh che richiede una parte di codice aggiuntiva per la selezione delle celle di interfaccia tra le due mesh.

L'implementazione dei modelli *InterFoam* e *overInterDyMFoam* è sviluppata all'interno delle *classi* disponibili del linguaggio *C++*.

Il *main code* per il solutore *interFoam* è rappresentato dallo pseudo-algoritmo seguente (tratto da [7]):

```
#1    Dichiarazione delle classi necessarie;
#2    Acquisizione modello di turbolenza impostato;

#3.0 Ciclo WHILE

#3.1  Acquisizione e calcolo del timestep impostato e dei limiti
      su numero di Courant;

      #3.2.0 Ciclo WHILE
      #3.2.1 Calcolo ed aggiornamento sulla posizione
              della mesh e algoritmo per la soluzione
              del metodo Inverse Distance;

      #3.2.2 Calcolo dei flussi sui bordi delle celle;

      #3.2.3 Soluzione dell equazione della
              Volume Fraction tramite le
              librerie "alphaControls.H" e
              "alphaEqnSubCycle.H";

      #3.2.4 Soluzione dell equazione della quantita di moto
              per il calcolo della velocita di guess tramite
```

```

        la libreria "UEqn.H";

    #3.2.5  Correzione della pressione di guess e soluzione
           dell'equazione della pressione dell' algoritmo
           PIMPLE tramite la libreria "pEqn.H";

    #3.2.6  Calcolo della turbolenza tramite il modello
           scelto e correzione dei parametri in funzione
           della turbolenza calcolata;

    #Fine ciclo 3.2.0

#Fine del ciclo 3.1

```

Nella parte iniziale vengono dichiarate le classi che il codice utilizza. All'interno di ogni classe sono contenute le informazioni per la creazione degli oggetti e le funzioni disponibili per gli oggetti appartenenti alla determinata classe. Notevole importanza ha la classe *createFields.H* che assegna i valori di campo alla griglia di calcolo costruita, la classe *pimpleControl.H* che contiene le funzioni per la gestione e la soluzione tramite l'algoritmo PIMPLE. Altre classi importate dichiarate sono le classi inerenti alla numerica dei volumi finiti come le classi: *fvCFD.H*, *fvOptions.H*, *EulerDdtScheme.H*, *localEulerDdtScheme.H*, *CrankNicolsonDdtScheme.H* presenti nella cartella del software */src*.

Le operazioni vere e proprie del codice hanno inizio alla riga *3.0*, con l'instaurazione del primo ciclo *while*. Questo ciclo è controllato dalla funzione *runTime.run()*, una funzione che verifica che il tempo della simulazione sia minore del tempo impostato come tempo limite. Le prime operazioni di questo ciclo sono riguardanti la movimentazione della mesh, e del calcolo del numero di Courant e dei flussi ai bordi delle celle. Queste sono fasi preparative per il calcolo delle grandezze tramite soluzione di NavierStokes.

Successivamente, vengono invocate due classi importanti: la classe *alphaControls.H* e la classe *alphaEqnSubCycle.H*. Le entità richiamate aggiungono al codice l'algoritmo di soluzione dell'equazione della frazione in volume per la finalizzazione del metodo Volume of Fluid per la soluzione del flusso bifase. I passaggi interni dell'algoritmo sono: acquisizione dei dati relativi al numero di cicli, inizializzazione delle variabili, esecuzione del *gamma subCycle* tramite un oggetto, all'interno del quale viene risolta l'equazione della frazione in volume. Alla fine del ciclo, vengono aggiornate sia la curvatura  $\kappa$  sia la densità dei fluidi.

Dopo l'interrogazione delle librerie per il flusso bifase, viene svolto il primo

passaggio relativo alle equazioni di Navier-Stockes.

Il ciclo richiama la classe *pimpleControl.H* che implementa le istruzioni per il calcolo attraverso l’algoritmo descritto nella sezione precedente.

Il primo step, svolto dalla classe *UEqn.H*, ha lo scopo di risolvere l’equazione della quantità di moto per il calcolo delle velocità associate alla pressione di *guess*. La parte di file adatta alla soluzione dell’equazione è:

```
fvVectorMatrix UEqn
(
    fvm::ddt(rho, U) + fvm::div(rhoPhi, U)
    + [...]
    + [...]
    ==
    fvOptions(rho, U)
);
```

Il passo successivo, rappresentato dal ciclo *while*, implementa i passaggi di correzione della pressione tramite la classe *pEqn.H*.

```
while (pimple.correctNonOrthogonal())
{
    fvScalarMatrix p_rghEqn
    (
        fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
    );

    p_rghEqn.setReference(pRefCell, ...
        ...getRefCellValue(p_rgh, pRefCell));

    p_rghEqn.solve(mesh.solver(...
        ...p_rgh.select(pimple.finalInnerIter())));

    if (pimple.finalNonOrthogonalIter())
    {
        [...]
    }
}
```

Chiude il ciclo globale l’implementazione del codice per la modellazione della turbolenza.

La parte inerente alla dinamica del corpo rigido tramite *oversteMesh* è contenuta nel codice *overInterDyMfoam*. Il codice è un’estensione dell’algoritmo *interFoam* che viene opportunamente modificato in alcune delle sue parti per, appunto, accogliere la logica di *oversetMesh* al suo interno.

I passaggi logici che si aggiungono per la soluzione della dinamica attraverso overset mesh sono:

1. Assegnazione del *cellType* alle celle del dominio di calcolo;
2. Interpolazione dei dati all'interfaccia tra overset e background;
3. Implementazione della libreria per la dinamica di corpo rigido;
4. Movimento della mesh di overset.

Riassumendo, graficamente, il ciclo di operazioni logiche che vengono eseguite dai codici *interfoam* e *overInterDyMFoam* verte le seguenti fasi:

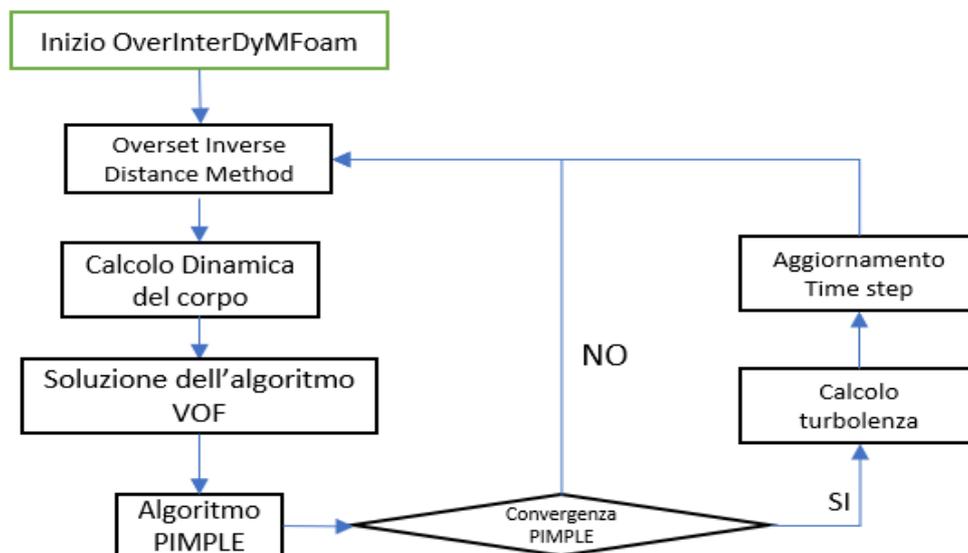


Figure 4.2: Pseudo-algoritmo OverInterDyMFoam [4]

## 4.3 Paraview

La fase finale della simulazione, il post-processing, si svolge con un altro software open-source, *Paraview*.

Paraview è un software che, come OpenFoam, viene interpretato come *multi-purpose*, infatti il suo utilizzo non è specifico per un ambito preciso, può essere utilizzato per l'elaborazione dei dati per una moltitudine di situazioni. Questo software è sviluppato per l'analisi di una grande quantità di dati.

Nell'ambito dell'utilizzo in accoppiamento con OpenFoam, Paraview acquisisce i dati delle cartelle *Processors-n* oppure delle cartelle temporali in funzione del fatto si voglia, rispettivamente, ricostruire il dominio nelle condizioni in cui viene parallelizzato o ricostruito.

Nell'ambiente di calcolo e di elaborazione dei dati, in Paraview, i dati della simulazione possono essere, elaborati per la visualizzazione di entità vettoriali, stramlines, colorazioni ad-hoc e tutto il necessario per la visualizzazione dei fenomeni fluidodinamici e numerici della simulazione.

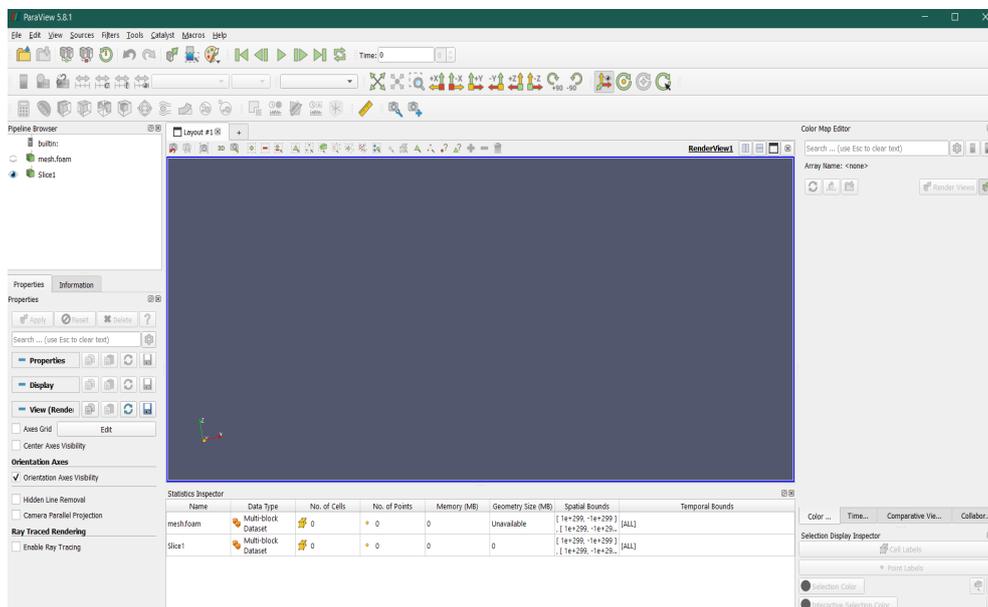


Figure 4.3: Interfaccia di Paraview

# Chapter 5

## Caso studio

*Nella preparazione di un codice di Computational Fluid Dynamics, a monte di qualsiasi settaggio del codice di calcolo, è indispensabile definire le esigenze di progetto e i dati che si vogliono ottenere. È necessario capire, quali grandezze sono utili al progetto per il quale si sta svolgendo la simulazione, a che livello di precisione è necessario calcolarli e quali sono le fenomenologie fisiche che si vogliono calcolare nel modello, dedicando ad essere il giusto costo computazionale; per esempio, quali forze sono necessarie, eventuali spostamenti del corpo, il grado di definizione della turbolenza presente ed altre.*

*Una volta definite queste considerazioni si svolge un'analisi sulla fattibilità degli studi in funzione dei mezzi che si hanno a disposizione, infatti, la CFD è un processo che può diventare anche molto costoso in termini computazionali fino a diventare impraticabile ai fini del progetto. È necessario operare tramite delle semplificazioni per riuscire a costruire delle simulazioni che riescano a calcolare tutto quello che è necessario ma con limitate richieste in termini di costi computazionali.*

*Nell'ambito delle imbarcazioni a vela i motivi per cui si ricorre alla CFD sono molteplici: si possono studiare le forze sulle vele, l'interazione aerodinamica tra vele e la strumentazione sulla coperta, oppure, si possono studiare i componenti immersi come appendici e/o scafo considerando l'assetto che la barca assume in varie condizioni di navigazione, fondamentale per comprendere al meglio le prestazioni del mezzo.*

*Il seguente capitolo è dedicato alla definizione del caso studio e alla descrizione dei settaggi e delle logiche utilizzate per le simulazioni svolte con il software OpenFoam. L'obiettivo delle simulazioni è calcolare il funzionamento della barca nel suo complesso mediante disponibilità minime in termini di software, OpenSource, ed Hardware, di tipo domestico.*

## 5.1 Obiettivi della simulazione CFD

Le necessità progettuali di una barca da regata ad alte prestazioni impongono l'ottimizzazione delle diverse andature possibili. Calcolare in fase di design le forze agenti sulla barca, in particolare la resistenza all'avanzamento, l'assetto che essa assume, ed il regime di navigazione, se dislocante o planante è una fase ambita da tutti i progettisti navali. La complessità delle richieste è proporzionale alla complessità del codice di calcolo implementato e alle risorse necessarie, per questo, nella maggior parte dei casi vengono utilizzate delle metodologie approssimate ma economiche.

L'obiettivo delle simulazioni descritte nel seguito è quello di ricreare la situazione di navigazione considerando tutti i componenti della barca limitando il costo computazionale e la complessità dell'intero sistema di calcolo. Si tiene in considerazione, infatti, l'effetto fluidodinamico di deriva e timone e delle interazioni tra di essi e con la superficie dello scafo. Inoltre, si introducono nel dominio di calcolo le forze delle vele, con lo scopo di completare lo stato di equilibrio anche con le forze ed i momenti prodotti da queste entità fondamentali per il funzionamento della barca a vela. La caratteristica fondamentale delle simulazioni implementate consiste nella possibilità di calcolare lo stato di equilibrio tenendo in considerazione ben tre gradi di libertà, quelli semi-indipendenti dall'equipaggio. I DoFs calcolati sono:

- Traslazionale lungo  $z$ , traslazione verticale;
- Rotazionale lungo  $y$ . sbandamento;
- Rotazionale lungo  $x$ , trim;

Gli input fisici della simulazione, ottenuti dalle esigenze di progetto e dalle caratteristiche di funzionamento della barca di categoria MELGES 24, sono rappresentati dai seguenti parametri:

- La velocità di avanzamento della barca, la componente lungo la direzione del moto;
- La velocità di scarroccio, che è la componente normale alla direzione di avanzamento della barca;
- Posizione sulla barca dell'equipaggio in funzione del regime di navigazione;
- Masse caratteristiche della barca, del bulbo, e dell'equipaggio.

- Forze e momenti generati dalle vele e dalle appendici calcolate da simulazioni ad-hoc.

Gli output fisici della simulazione riguardano la condizione di equilibrio che raggiunge la barca in una determinata condizione di navigazione:

- Le forze agenti sulle geometrie immerse:
  - Resistenza;
  - Forza laterale;
  - Forza verticale: che serve più per capire se la simulazione produce un risultato sensato, perchè la forza verticale deve coincidere, a meno di un certo errore, con il peso della barca.
- Componenti di assetto nello stato di equilibrio:
  - Traslazione verticale, lungo  $z$ ;
  - Rotazione di trim, lungo  $y$ ;
  - Rotazione di sbandamento, lungo  $x$ .
- Forma del sistema di onde generato nella condizione di equilibrio;
- Distribuzione di pressione sulla carena.

Spesso, durante la progettazione navale, è utile osservare il comportamento della carena, isolata dagli altri componenti ed in una condizione di navigazione simmetrica, questo per studiare le caratteristiche di navigazione per ottimizzare le geometrie. Per effettuare questo studio, più semplice del precedente, le simulazioni complete descritte sopra vengono modificate e semplificate per effettuare questo tipo di calcolo. In questo caso gli input della simulazione sono:

- La velocità di avanzamento della barca, allineata con l'asse longitudinale della barca;
- Masse e momenti di inerzia caratteristici della carena.

Mentre gli output provenienti dal sistema di calcolo sono:

- Le forze agenti sulle geometrie immerse:
  - Resistenza;

- Forza laterale;
- Forza verticale: anche in questo caso è utile a comprendere il grado di equilibrio raggiunto;
- Componenti di assetto nello stato di equilibrio:
  - Traslazione verticale, lungo  $z$ ;
  - Rotazione di trim, lungo  $y$ .
- Forma del sistema di onde generato nella condizione di equilibrio.

Nel seguito vengono descritte entrambe le simulazioni, partendo dal modello completo fino alla sua semplificazione per lo studio simmetrico.

## 5.2 Struttura della simulazione

La simulazione per il calcolo delle caratteristiche di navigazione del MELGES 24 è strutturata come segue.

La velocità relativa tra barca e fluido è generata dal flusso fluido inserito nelle condizioni al contorno: le superfici frontali, grazie alla condizione di inlet, consentono l'ingresso del fluido nel dominio e quelle posteriori, con una condizione di outlet, sono adattate alla fuoriuscita dello stesso dal dominio. La carena è posizionata all'interno del dominio stesso ed è libera di muoversi solo sui gradi di libertà corrispondenti alle incognite, mentre è bloccata, numericamente, sugli altri gradi di movimento. Le forze delle componenti presenti nella realtà, che intervengono nella condizione di equilibrio e generate da componenti che non sono contenute nel modello della carena, vengono inserite come forze esterne applicate nel baricentro della carena. In particolare, per completare lo stato di equilibrio, vengono calcolati i momenti generati da appendici e vele rispetto al baricentro per mezzo di simulazioni costruite ad-hoc. Gli effetti, successivamente vengono aggiunti alla dinamica della carena come entità esterne.

Strutturando la simulazione come descritto si agevola il calcolo in quanto il sistema *barca con appendici e vele* viene studiato separatamente e le criticità dello studio delle varie entità non converge in un'unica simulazione. Inoltre, la fase di *processing* delle diverse simulazioni può essere richiamato-contemporaneamente in calcolatori anche diversi.

Il movimento della carena, all'interno della simulazione, viene ottenuto con la logica di *OversetMesh*. Questo aspetto è dato dal fatto che non si

conosce lo stato di equilibrio finale della barca che, può assumere un orientamento anche molto diverso dalla condizione di posizionamento iniziale del corpo all'interno della simulazione. Al fine di evitare le problematiche di distorsione eccessiva della griglia di calcolo dovuta ad un grande movimento della carena, si opta per una logica di *oversetMesh* la quale, correttamente settata, permette anche grandi discostamenti dalla condizione iniziale del corpo.

### 5.2.1 Dominio di calcolo

Le componenti che costituiscono il dominio di calcolo sono due, la prima è il "contenitore" esterno per limitare l'estensione dell'ambiente che circonda la carena. Questo viene chiamato *background*.

Il secondo elemento importante che compone il dominio, è l'oggetto (in questo caso la carena), di cui si vogliono studiare le caratteristiche di navigazione. La carena viene posta in una posizione ben definita rispetto alle *boundaries* del *background* per evitare che le condizioni al contorno influiscano sul comportamento del fluido attorno alla barca. Inoltre per poter utilizzare l'approccio di *OversetMesh* quindi viene predisposto un blocco a forma di parallelepipedo per contenere la geometria mobile, che si muoverà in maniera solidale con essa. Le dimensioni dei due blocchi sono le seguenti:

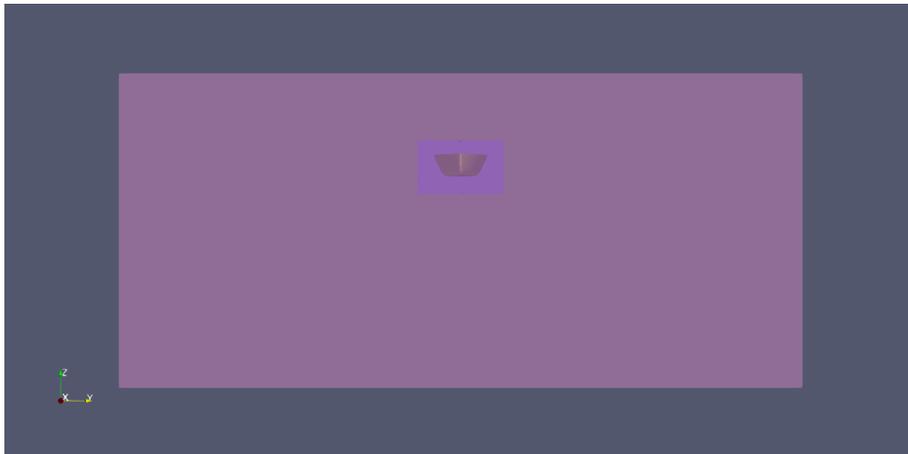
Sottodominio	lunghezza (x) [m]	larghezza (y) [m]	altezza (z) [m]
background	56.0	32.0	14.6
overset	10.0	4.0	2.5

Table 5.1: Dimensioni del dominio

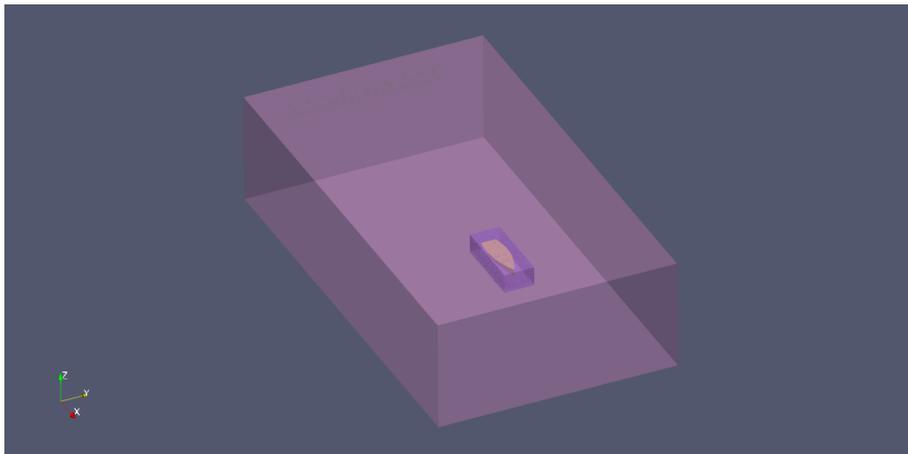
Tali dimensioni, per quanto riguarda il *background* vengono scelte in funzione dell'influenza delle *boundaries* sulla geometria, quindi le superfici di contorno devono essere abbastanza lontane dalla carena ma contemporaneamente devono essere tali da non richiedere un costo computazionale eccessivo. Nella zona posteriore della barca, data la presenza dell'onda generata, lo spazio lasciato tra poppa e superficie di uscita del fluido equivale alla lunghezza di 8 barche. Per quanto riguarda *l'overset* lo spazio in eccesso rispetto alla carena viene scelto in base al numero di celle presenti tra le superfici e il corpo, che dev'essere più alto possibile, e la distanza effettiva che, se piccola garantisce uno spostamento più basso delle superfici, garanzia di maggiore stabilità del codice di calcolo.



(a) Vista laterale del dominio



(b) Vista frontale del dominio



(c) Vista del dominio in prospettiva

Figure 5.1: Dominio del codice di calcolo

## 5.2.2 Costruzione e dimensionamento della mesh

La discretizzazione spaziale del dominio, la generazione della *mesh*, avviene tramite le *Utilities* di OpenFoam: *BlockMesh* e *SnappyHexMesh* descritte sopra, che vengono richiamate attraverso le *Dictionaries* *BlockMeshDict* e *SnappyHexMeshDict*.

I livelli di discretizzazione utilizzati sono riportati in tabella:

Background mesh			
Livello	Ambiente	Dimensione (x,y) [m]	AR
Livello 0	background	0.8	1
Livello 0	wave coarse 1	0.8	0.5
Raffinamento xyz	wave coarse 2	0.4	0.25
Raffinamento xy	wave fine 1	0.2	0.5
Raffinamento xyz	wave fine 2	0.1	0.5
Raffinamento xyz	overset refinement 1	0.2	1
Raffinamento xyz	overset refinement 1	0.1	1

Table 5.2: Dimensione della mesh di background

Overset Mesh			
Livello	Ambiente	Dimensione (x,y) [m]	AR
Livello 0	Overset	0.08	1
Raffinamento superficie	minima	Livello 2 (0.02)	-
Raffinamento superficie	massima	Livello 1 (0.04)	-

Table 5.3: Dimensione della mesh di overset

La discretizzazione del *background* avviene con la *dictionary* *blockMeshDict* e la dimensione delle celle create viene associata al *livello 0*. Nel *background* vengono inseriti alcuni raffinamenti per la discretizzazione dell'onda che si viene a formare dietro la carena e per l'alloggiamento dell'overset. Le celle che si sovrappongono tra *background* ed *overset* in quanto, per una migliore interpolazione dei dati tra le due entità del dominio, è necessario che la dimensione delle celle *donors* siano della stessa grandezza delle celle *acceptors*.

Ulteriore aspetto importante nel dimensionamento delle celle è l'*Aspect Ratio*, il rapporto tra l'altezza e la larghezza/profondità della cella.

L'ottimizzazione del costo computazionale e della precisione nella soluzione delle equazioni prevede che l'onda generata dalla barca debba essere discretizzata con celle con  $AR = 2$  o  $AR = 4$  in funzione del fatto che l'onda sia più o meno ripida.

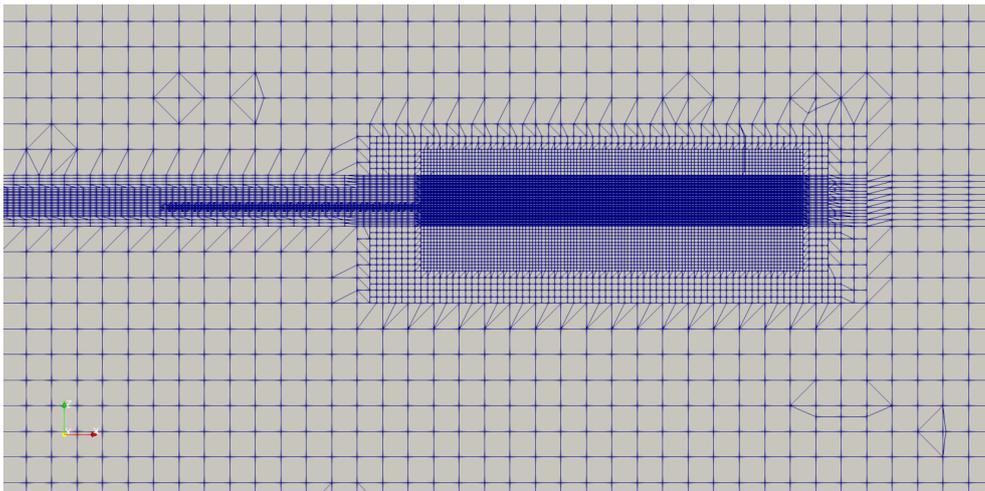
Un altro requisito essenziale che viene tenuto in considerazione nello sviluppo della discretizzazione è la forma delle celle di interfaccia tra *background* e *overset*. In questo caso le celle devono essere della stessa grandezza e l'AR delle celle di *overset* devono rispettare il più possibile la condizione di  $AR = 1$  per un migliore funzionamento dell'algoritmo *snappyHexMesh*.

Durante la progettazione della griglia di *overset*, quindi, si parte con il livello 0 corrispondente alla dimensione del raffinamento rispettivo nel *background*.

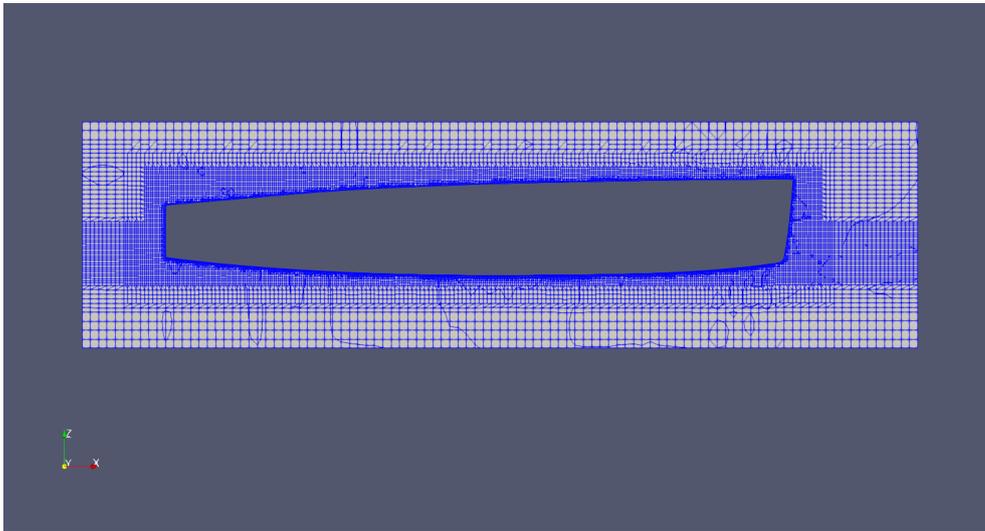
I raffinamenti applicati all'*overset* sono tali da mantenere le celle attorno allo scafo di dimensione cubica e tali da discretizzare l'onda con celle di AR pari a 4 essendo un'onda *smooth*. L'altezza del raffinamento per l'onda in numero di celle in altezza è tale da contenere l'altezza dell'onda stessa in almeno 10 celle.

In ultimo, le celle sulla superficie della carena, per un rispetto ottimale del contorno della stessa superficie sono di livello da 2 a 4 con raffinamento al livello 4 laddove la superficie crea un angolo minore di 20 gradi.

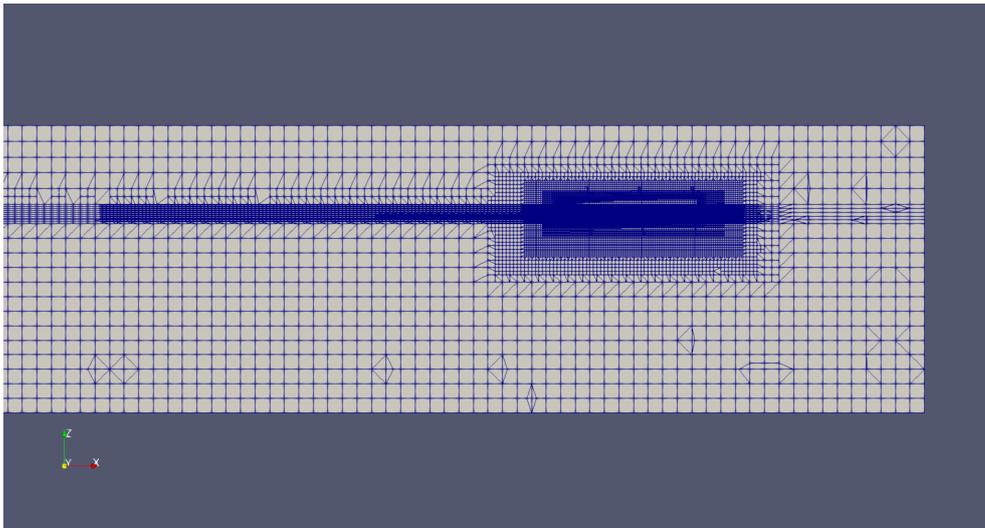
Il risultato delle due mesh è riportato in figura 5.3.



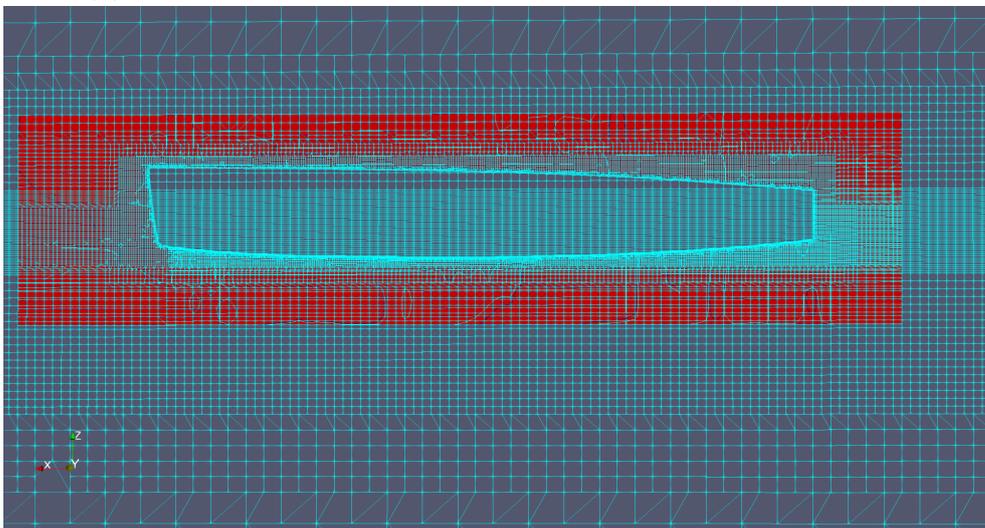
(a) Discretizzazione del dominio di background



(a) Discretizzazione del dominio di overset



(b) Discretizzazione dominio globale



(c) Sovrapposizione della mesh di background (azzurra) e di overset (rossa)

Figure 5.3: Discretizzazione dominio spaziale

## Overset Mesh

L'implementazione della logica di *OversetMesh* avviene tramite la creazione dei due singoli blocchi del dominio e successiva sovrapposizione degli stessi.

L'attuazione dell'approccio *oversetMesh* per la dinamica del corpo rigido, per quanto riguarda la sola generazione della mesh globale, segue lo pseudo-algoritmo di seguito:

1. Generazione dell'Overset:
  - (a) *Block mesh* per la costruzione del blocco di overset;
  - (b) *SnappyHexMesh* per inserire il CAD nel blocco dell'overset;
2. Generazione del Tank:
  - (a) *BlockMesh* per il tank;
  - (b) *refineMesh* per effettuare tutti i raffinamenti descritti;
3. Unione dell'overset e del Tank con la utility *mergeMeshes*.

## Scelta del Prism Layer

La discretizzazione dello strato limite tramite le celle prismatiche è fondamentale per un corretto calcolo delle forze agenti sulla superficie ed una migliore convergenza degli algoritmi dedicati ai modelli per il calcolo della turbolenza. I parametri disponibili nel tool *snappyHexMesh* sono i seguenti:

- Numero di strati;
- Spessore strato esterno;
- Rapporto di crescita tra uno strato ed il suo contiguo.

Per poter assegnare ad ognuno dei parametri il valore corretto, è necessario fare dei calcoli preliminari.

Lo spessore del primo layer influenza il valore del parametro adimensionale  $y^+$ ; per l'implementazione del modello di turbolenza  $k - \epsilon$  è consigliabile che il valore di *wall*  $y^+ > 30$  quindi l'intero spessore di *viscous sublayer* e *buffer layer* è compreso nella prima cella sulla superficie del corpo. Per il calcolo di questo parametro si parte dal *wally+*, dalle caratteristiche del fluido, del flusso e del corpo.

$$\begin{aligned}
 U^+ &= \frac{1}{k} \ln y + C \\
 U^+ &= \frac{U}{U_\tau} \Rightarrow U_\tau = \frac{U}{U^+} \\
 y^+ &= \frac{y u_\tau}{\nu} \Rightarrow y = \frac{y^+ \nu}{u_\tau}
 \end{aligned}
 \tag{5.1}$$

Una stima sullo spessore globale dello strato limite è, inoltre, necessaria alla definizione del numero degli strati e del fattore di crescita tra gli strati. Tra le due solo una è una variabile indipendente, il numero di strati, per questioni di costo computazionale è impostato ad un valore intorno ai 5 e di conseguenza, unendo al sistema lo spessore totale, lo spessore iniziale, si arriva alla verifica dello strato limite. Con i parametri descritti la discretizzazione dello strato limite appare come in figura 5.4.

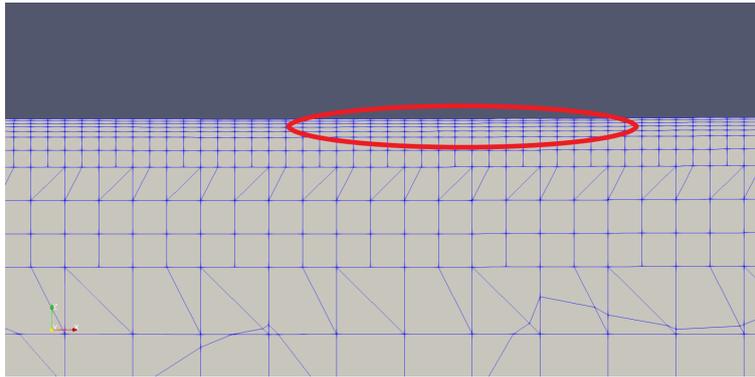


Figure 5.4: Discretizzazione del Prism Layer sulla superficie dello scafo

### 5.3 Condizioni al contorno

Le superfici adatte per l'impostazione delle condizioni al contorno sono le superfici corrispondenti ai bordi del *tank*, dell'*overset*, e del corpo in esame, le superfici in corrispondenza delle quali alcuni valori di campo, di velocità o di pressione, sono noti anche prima del calcolo. In particolare, due superfici del tank devono necessariamente rappresentare l'ingresso del flusso bifase di aria e acqua, due superfici vengono dedicate all'uscita del fluido e sulla superficie del corpo si impone la condizione no slip condition, impostando il valore di velocità sulle facce corrispondenti pari a zero. Le restanti superfici di contorno, sono di altro tipo: di simmetria, wall.

Le due condizioni di ingresso e le corrispondenti due di uscita sono necessarie dal momento che la velocità di simulazione presenta due componenti, una di avanzamento ed una corrispondente alla velocità di scarroccio. In ragione della presente considerazione si può dimostrare che le superfici laterali sono dedicate al flusso dovuto alla velocità di scarroccio mentre le superfici di background frontale e posteriore rappresentano entrata ed uscita del flusso lungo l'asse di avanzamento della barca.

Nelle pagine successive vengono descritte le condizioni al contorno utilizzate nel modello completo della carena. In OpenFoam le *boundary conditions* vengono impostate creando, nella cartella  $\theta$ , i file relativi alla grandezza di cui si vogliono descrivere le caratteristiche su tutte le superfici adatte. In funzione del solutore scelto, le grandezze da descrivere variano; per l'algoritmo *overInterDyMFoam* le dimensioni richieste per il settaggio delle boundary conditions sono: velocità, pressione specifica (dato che nel solutore viene considerata l'effetto idrostatico dell'acqua), indice della zona in funzione del tipo di mesh, se di background o di overset, pointDisplacement, energia cinetica turbolenta, rateo di dissipazione della turbolenza, turbulent viscosity.

### 5.3.1 Velocità, U

La velocità del fluido è un dato di *input* che viene determinato dalle esigenze del progetto. Nel caso studio la velocità di test per le simulazioni viene stabilita con il Velocity Prediction Program. La velocità della fase liquida e aeriforme è la stessa in modulo, direzione e verso.

Boundary	Tipo	Valore	Descrizione
inlet	<i>fixedValue</i>	<i>uniform</i> ( $U_x, U_y, U_z$ )	Valore vettoriale costante ed uniforme rispetto alle celle della superficie nella forma <i>uniform</i> ( $U_x, U_y, U_z$ )
side_in	<i>fixedValue</i>	<i>uniform</i> ( $U_x, U_y, U_z$ )	-
outlet	<i>zeroGradient</i>	-	La condizione di gradiente nullo impone la condizione di gradiente, in questo caso della velocità, pari a zero; del tipo $\frac{dU}{dt} = 0$
side_out	<i>zeroGradient</i>	-	-
bottom	<i>zeroGradient</i>	-	-
side_out	<i>zeroGradient</i>	-	-
atmosphere	<i>pressureInletOutletVelocity</i>	tangentialVelocity \$internal-Field	Condizione di velocità per superfici <i>inlet/outlet</i> con pressione specificata.
hull	<i>movingWallVelocity</i>	uniform (0 0 0)	La condizione impone la velocità nulla sulla superficie dello scafo
overset	<i>overset</i>	-	La condizione è dedicata alla definizione delle superfici di interfaccia dell' <i>overset</i> sulle quali si trovano le celle adatte all'interpolazione dei dati

Table 5.4: Condizioni al contorno per la velocità

### 5.3.2 Pressione specifica, $p_{rgh}$

La condizione di pressione è un dato ignoto a priori e si stabiliscono le condizioni al contorno in maniera tale che le pressioni presenti sulle *boundaries* garantiscano un flusso con la velocità imposta nelle condizioni di velocità. All'uscita del flusso si impone che la pressione abbia gradiente nullo per non influenzare il flusso.

Boundary	Tipo	Valore	Descrizione
inlet	<i>fixedFluxPressure</i>		Gradiente di pressione adeguato affinché la condizione di velocità venga rispettata
side_in	<i>fixedFluxPressure</i>		-
outlet	<i>zeroGradient</i>	-	La condizione imposta la condizione di Neumann di gradiente di pressione nullo sulla superficie, $\nabla p = 0$
side_out	<i>zeroGradient</i>	-	-
bottom	<i>zeroGradient</i>	-	-
atmosphere	<i>totalPressure</i>	uniform 0	Prevede l'indicazione della pressione totale dell'aria
hull	<i>fixedFluxPressure</i>	-	-
overset	<i>fixedFluxPressure</i>	-	-

Table 5.5: Condizioni al contorno per la pressione specifica

### 5.3.3 ZONE ID

Definizione delle zone per la logica *Overset Mesh*.

boundary	tipo	valore	descrizione
inlet	<i>zeroGradient</i>		Condizione di Neumann sulla grandezza corrispondente
side_in	<i>zeroGradient</i>		-
outlet	<i>zeroGradient</i>		-
side_out	<i>zeroGradient</i>		-
bottom	<i>zeroGradient</i>		-
atmosphere	<i>zeroGradient</i>		-
hull	<i>zeroGradient</i>		-
overset	<i>overset</i>	<i>uniform0</i>	Condizione per l'interpolazione

Table 5.6: Condizione al contorno per l'indice della zona, ZONE ID

### 5.3.4 PointDisplacement

Definizione della posizione iniziale dei punti.

boundary	tipo	valore	descrizione
inlet	<i>fixedValue</i>	<i>uniform(0 0 0)</i>	Condizione di Dirichlet
side_in	<i>fixedValue</i>	<i>uniform(0 0 0)</i>	-
outlet	<i>fixedValue</i>	<i>uniform(0 0 0)</i>	-
side_out	<i>fixedValue</i>	<i>uniform(0 0 0)</i>	-
bottom	<i>zeroGradient</i>	-	Condizione di Neumann
atmosphere	<i>fixedValue</i>	<i>uniform(0 0 0)</i>	-
hull	<i>calculated</i>	-	Condizione calcolata sulla superficie
overset	<i>zeroGradient</i>	-	-

Table 5.7: Condizioni al contorno per il *Point Displacement*

### 5.3.5 Energia cinetica turbolenta, $k$

Valori iniziali per le grandezze del codice di calcolo per la turbolenza. Vengono settati i valori di *default* che valgono nel caso di un flusso bisfa ed incomprimibile.

boundary	tipo	valore	descrizione
inlet	<i>fixedValue</i>	<i>uniform 0.01</i>	Condizione di Dirichlet
side_in	<i>fixedValue</i>	<i>uniform 0.01</i>	-
outlet	<i>inletOutlet</i>	<i>uniform 0.01</i>	Condizione che specifica lo stato di outlet del fluido, con indicazione di inlet nel caso di ricircolo
side_out	<i>inletOutlet</i>	<i>uniform 0.01</i>	-
atmosphere	<i>inletOutlet</i>	<i>uniform 0.01</i>	-
hull	<i>kqRWallFunction</i>	<i>uniform 0.01</i>	Condizione di definizione delle caratteristiche di parete per il calcolo della turbolenza
overset	<i>overset</i>	<i>uniform 1</i>	Condizione per l'interpolaizione con logica overset mesh

Table 5.8: Condizioni al contorno per l'energia cinetica turbolenta

### 5.3.6 Rateo di dissipazione della turbolenza, $\epsilon$

Come per il caso dell'energia cinetica turbolenta, il valore di questa grandezza viene definito con le quantità di *default* dell'algoritmo adatte al caso studio.

boundary	tipo	valore	descrizione
inlet	<i>inletOutlet</i>	<i>uniform0.1</i>	Condizione che specifica lo stato di outlet del fluido, con indicazione di inlet nel caso di ricircolo
side_in	<i>inletOutlet</i>	<i>uniform0.1</i>	-
outlet	<i>inletOutlet</i>	<i>uniform0.1</i>	-
side_out	<i>inletOutlet</i>	<i>uniform0.1</i>	-
atmosphere	<i>inletOutlet</i>	<i>uniform0.1</i>	-
hull	<i>epsilonWallFunction</i>	$C_{\mu} = 0.09$ $kappa = 0.41$ $E = 9.8$	Definizione dei coefficienti utili al calcolo della turbolenza con modello $k-\epsilon$
overset	<i>overset</i>	<i>uniform 1</i>	Condizione per l'interpolaizione con logica overset mesh

Table 5.9: Condizioni al contorno per il rateo di dissipazione della turbolenza

### 5.3.7 Turbolent viscosity, $\tilde{\nu}$

boundary	tipo	valore	descrizione
inlet	<i>fixedValue</i>	<i>uniform 0</i>	Condizione di Dirichlet
side_in	<i>fixedValue</i>	<i>uniform 0</i>	-
outlet	<i>zeroGradient</i>	-	Condizione di Neuman
side_out	<i>zeroGradient</i>	-	-
atmosphere	<i>calculated</i>	<i>uniform 0</i>	Valori calcolati sulla superficie
hull	<i>nutKWallFunction</i>	$C_{\mu} = 0.09$ $kappa = 0.41$ $E = 9.8$	Definizione dei coefficienti utili al calcolo della turbolenza nello strato limite a parete
overset	<i>overset</i>	<i>uniform 1</i>	Indicazione della condizione di interpolazione

Table 5.10: Condizioni al contorno per il *Point Displacement*

## 5.4 Il codice principale: `controlDict`, `fvSolution`, `fvScheme`

L'applicazione utilizzata è l'applicazione *overInterDyMFoam*, che permette di risolvere le equazioni di Navier-Stokes per un flusso bifase incomprimibile, immiscibile implementando la logica dell'overset mesh come studiato nel capitolo 4.

Il solutore viene richiamato attraverso la dictionary *controlDict* e nei documenti *fvScheme* e *fvSolution*, tutti collocati nella cartella *system* della simulazione in openFoam.

### 5.4.1 `controlDict`

Questa dictionary richiama ed avvia il solutore stabilito per la simulazione. Inoltre, comprende la dichiarazione di ulteriori librerie necessarie e settaggi per il controllo della numerica e per la gestione della simulazione. La descrizione della *Dictionary* è riportata in tabella 5.11.

Le librerie utilizzate vengono indicate all'inizio dell'oggetto sono:

I settaggi impostati per le librerie descritte sono riportati nella tabella 5.11.

### 5.4.2 `fvSolution`

La gestione delle caratteristiche numeriche degli algoritmi che risolvono le equazioni per il caso studio sono contenute nel foglio *fvSolution*. Per la simulazione in esame vengono settate le tolleranze per le grandezze incoognite delle equazioni:

Grandezza	Residuo
U	$10^{-8}$
p	$10^{-10}$
cellDisplacement	$10^{-8}$
$\kappa$	$10^{-8}$
$\epsilon$	$10^{-8}$

Table 5.12: Tolleranza sui residui per le simulazioni della carena

Le caratteristiche e le tolleranze per il codice per la soluzione dell'equazione bifase in tabella 5.13:

Ingresso	Valore	Descrizione
Librerie aggiunte	<i>overset rigidBody-Dynamics fvMotion-Solvers</i>	
Application	overInterFoam	Indicazione del solutore da utilizzare per la simulazione.
startTime	0.0	Istante temporale iniziale. Dopo l'interruzione della simulazione assume il valore dell'ultimo salvataggio.
endTime	15.0	Istante temporale finale. Quando la simulazione raggiunge l'istante temporale corrispondente, il calcolo si interrompe
deltaT	$\frac{C_{o_{int}} \cdot \Delta x}{v}$	Il passo di discretizzazione temporale standard è calcolato per mantenere un <i>Courant</i> all'interfaccia tra <i>overset</i> e <i>background</i> pari a $C_{o_{int}} = 0.5$ .
runTimeModificable	<i>yes</i>	Possibilità di modificare il <i>timeStep</i> durante il calcolo.
adjustTimeStep	<i>yes</i>	Se attivata, modifica il <i>timeStep</i> affinché mantenga un numero di <i>Courant</i> massimo al disotto di un certo valore.
maxCo	3.0	Valore del numero di <i>Courant</i> massimo per il passo temporale variabile
maxAlphaCo	3.0	Valore del <i>Courant</i> massimo all'interfaccia tra i fluidi per il passo temporale variabile.
maxDeltaT	0.06	Valore di <i>timeStep</i> massimo raggiungibile dalla simulazione.

Table 5.11: Ingressi della dictionary *controlDict*

Grandezza	Residuo
nAlphacorr	4
nAlphaSubCycles	2
tolerance	$10^{-9}$

Table 5.13: Parametri per il controllo della soluzione dell'equazione della Volume Fraction per lo svolgimento delle simulazioni della carena

In ultimo, l'oggetto richiede in ingresso i settaggi per l'algoritmo PIMPLE. I dati modificati, rispetto alla configurazione di default sono:

Grandezza	Residuo
nCorrectors	3
nOuterCorrectors	1

Table 5.14: Parametri per il controllo dell'algoritmo PIMPLE

La scelta dei seguenti parametri ha importanti conseguenze sul costo computazionale; l'algoritmo utilizzato presenta un singolo *outerCorrector loop*, e tre *innerCorrectorLoops* di conseguenza, lo schema risolutivo implementato è lo schema PISO con un ciclo correttivo in più per migliorare ulteriormente il calcolo della pressione.

### 5.4.3 fvSchemes

Nell'oggetto *fvSchemes* sono contenute tutte le caratteristiche utilizzate per la discretizzazione delle equazioni e la soluzione numerica dei sistemi di equazioni. I valori di input utilizzati sono i settaggi di default del codice *overInterDyMFoam*.

## 5.5 Dinamica della carena, *dynamicMeshDict*

La gestione della dinamica del corpo di cui si vuole studiare l'interazione con il fluido è contenuta nell'oggetto *dynamicMeshDict*, nella cartella *constant*.

La prima parte della dictionary è dedicata alla definizione delle librerie disponibili nel software che si vogliono utilizzare ed ai parametri dei solutori che si scelgono, nonché ai parametri per la definizione della stabilità del codice. I parametri utilizzati nelle simulazioni sono riportati in tabella 5.15.

Le criticità riguardano la stabilità del codice, infatti, movimenti con accelerazioni eccessive del corpo creano instabilità numerica e divergenza della simulazione. La gestione della stabilità avviene tramite i coefficienti *accelerationRelaxation* che agisce, come visto, in maniera diretta sull'accelerazione dell'iterata corrente e l'*accelerationDamping* che, invece, agisce in maniera proporzionale all'accelerazione dell'iterata corrente. Durante l'esecuzione delle simulazioni il controllo su questi parametri permette di stabilizzare la simulazione inizialmente, gestire la velocità di movimento fino al punto di equilibrio nella parte più stabile della simulazione (circa dopo 1,5 secondi di

Ingresso	Valore	Descrizione
motionSolverLibs	rigidBodyMeshMotion	
dynamicFvMesh	dynamicOversetFvMesh	
motionSolver	rigidBodyMotion	
solver	type Newmark	
accelerationRelaxation	0.4	Il valore finale viene modificato da un valore molto basso al valore indicato per assicurare la stabilità del codice.
accelerationDamping	0.8	Lo smorzamento dell'accelerazione viene attivato per garantire la stabilità del calcolo.

Table 5.15: Parametri numerici di settaggio del codice per la dinamica del corpo in movimento

tempo fisico). Nella condizione di equilibrio finale, tuttavia, i valori dei due coefficienti devono essere abbastanza alti per non influenzare la condizione di equilibrio stessa, che una volta raggiunta garantisce accelerazioni e velocità circa nulle.

L'inserimento delle caratteristiche fisiche del corpo dichiarato prima nella mesh e poi nelle condizioni al contorno avviene nel foglio *dynamicMeshDict* attraverso il seguente script:

```
bodies
{
  hull
  {
    type          rigidBody;
    parent        root;

    mass          1196.0;  Massa della barca
    inertia       (1248.0 0 0 3274.0 0 3462.68);
                  //format (Ixx Ixy Ixz Iyy Iyz Izz)

    centreOfMass (0 0 0);
                  //Riferimento cartesiano

    transform     (1 0 0 0 1 0 0 0 1) (3.666 0 0.35);
                  //Centro di rotazione del corpo

    joint         //Gradi di liberta
    {
```

```

        type                composite;
        joints
        (
            {
                type Pz;
                //Traslazione verticale
            }

            {
                type Rx;
                //Rotazione su asse longitudinale
            }

            {
                type Ry;
                //Rotazione su asse trasversale
            }

        );
    }

    patches      ("hull.*");
                //Superfici costituenti il corpo
}

restraints //Vincoli applicati al corpo
{

    translationDamper //Smorzatore per la traslazione
    {
        type linearDamper;
        body hull;
        coeff 200000;
    }

        rotationDamper //Smorzatore per la rotazione
    {
        type sphericalAngularDamper;
        body hull;
        coeff 20000;
    }

    effetto momento delle appendici e delle vele
    {
        type externalForce;
        body hull;
        location      (3.666 0 -1.35);
        force      (320.0 -1427.0 0);
    }

    effetto forza di galleggiamento delle appendici
    {

```

```

        type externalForce;
        body hull;
        location      (3.666 0 -1.35);
        force      (320.0 -1427.0 0);
    }

    momento ribaltante del bulbo
    {
        type externalForce;
        body hull;
        location      (3.666 0 -1.35);
        force      (320.0 -1427.0 0);
    }

    momento ribaltante dell equipaggio
    {
        type externalForce;
        body hull;
        location      (3.666 0 -1.35);
        force      (320.0 -1427.0 0);
    }
}

```

All'interno del codice riportato, si introducono tutte le caratteristiche del corpo che interagisce con il fluido all'interno della simulazione.

La massa della barca, i momenti d'inerzia, e il baricentro vengono indicati nella prima parte. Il centro di rotazione viene inserito relativamente al centro di massa. Il punto di rotazione in input corrisponde al centro di massa della barca in assetto neutro.

Nella seconda parte vengono indicati i gradi di libertà, in questo caso, ritornando ai dati che si vogliono calcolare, si "liberano" : la traslazione verticale, la rotazione lungo l'asse longitudinale e la rotazione lungo l'asse trasversale.

Infine, nella terza ed ultima parte vengono inseriti i vincoli applicati al corpo. Gli smorzatori sono entità numeriche che vengono usati per rallentare in maniera proporzionale alla velocità il movimento del corpo. Siccome è necessario trovare una condizione stazionaria, gli effetti di inerzia della barca rallenterebbero inutilmente il raggiungimento di una posizione di equilibrio generando oscillazioni del corpo. Le altre forze aggiunte rappresentano l'effetto del bulbo e dell'equipaggio. L'effetto delle appendici e delle vele, calcolato nelle simulazioni ad-hoc.

### 5.5.1 Schema di funzionamento della barca nella simulazione

Inserire tutte le componenti che prendono parte nella dinamica della barca attraverso delle forze concentrate, disponibili in OpenFoam come *restraints*, richiede la schematizzazione delle forze realmente presenti sulla barca durante la particolare andatura. Le forze che si aggiungono al modello della barca sono:

**Effetto delle Appendici** Attraverso la resistenza e la portanza che si generano grazie alla componente di scarroccio, sviluppano un momento di sbandamento concondre a quello sviluppato dalle vele ed un momento che modifica l'angolo di trim portando la prua, nella maggior parte dei casi verso il basso. Le appendici si aggiungono alla dinamica della barca attraverso due entità di forza. La prima forza è la spinta idrostatica ottenuta dall'immersione dei corpi nell'acqua. Il secondo effetto riassume i momenti che le appendici applicano al centro di rotazione della carena. Questi momenti sono ricostruiti nella dictionary tramite delle forze applicate ad una distanza tale, dal centro di rotazione, da rappresentare il momento che le appendici sviluppano.

**Effetto delle Vele** Le vele, nella produzione della forza trainante e della sua componente trasversale generano un effetto analogo a quello delle appendici.

**Effetto del Bulbo** Il bulbo è una massa pari a  $m_b = 280Kg$  (nel caso del Melges 24), posizionata all'estremità della deriva, che, in funzione dello sbandamento produce un momento opposto all'angolo tanto maggiore quanto maggiore è lo sbandamento stesso. Viene inserita come forza con componente verticale concentrata nel centro di massa del bulbo. La tipologia di *restraint* dedicata per l'inserimento del bulbo, l'*externalForce*, garantisce lo spostamento del punto di applicazione in maniera solidale a quello del corpo principale mantenendo la direzione delle forza inalterata, questo comportamento rispecchia perfettamente ciò che succede nella realtà;

**Effetto dell'Equipaggio** L'equipaggio si posiziona sulla barca per cercare l'equilibrio laterale e longitudinale per la precisa andatura di navigazione. Come per l'effetto del bulbo, la forza che rappresenta l'equipaggio è la forza peso dello stesso applicata nel centro di massa dell'equipaggio durante la configurazione di navigazione. Sia il bulbo he l'equipaggio hanno la funzione di generare un momento ribaltante per limitare lo

sbandamento della barca durante le andature di bolina e traverso in particolare.

Gli schemi riepilogativi che riassumono lo schema barca inserito nella simulazione e inserito nella dictionary *dynamicMeshDict* è riportato nelle figure.

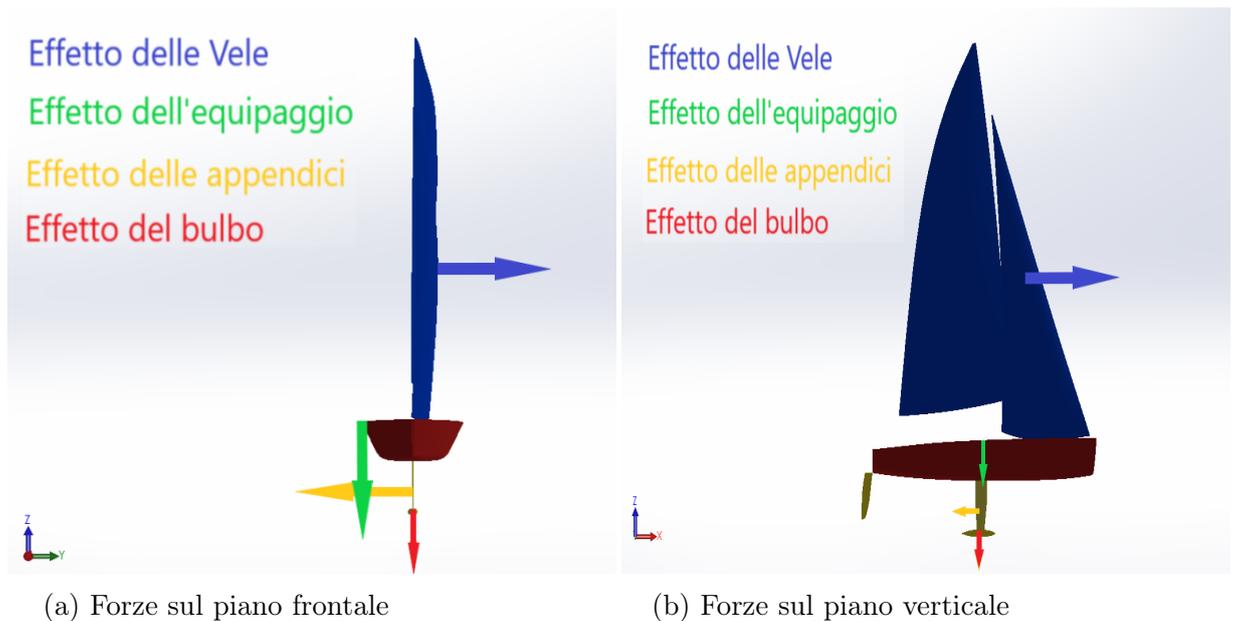


Figure 5.5: Schema delle forze esterne inserite nella dinamica della carena

## 5.6 Modello di Turbolenza

Il modello di turbolenza si imposta all'interno del foglio *turbulenceProperties* presente nella cartella */constant*. L'implementazione avviene tramite le righe seguenti:

```
simulationType      RAS;

RAS
{
  RASModel          kEpsilon;

  turbulence        on;

  printCoeffs       on;
}
```

Il settaggio delle costanti utilizzate dall'algoritmo, invece, vengono elencate negli oggetti delle condizioni al contorno dedicati alle grandezze della turbolenza. Per le simulazioni viene settato un modello  $k-\epsilon$ . La scelta è limitata dalle disponibilità computazionali, infatti, il modello scelto è un algoritmo che richiede un costo minore. L'alternativa è il modello  $k-\omega$ , migliore per le situazioni in cui si deve calcolare una forza su un oggetto immerso in un fluido. Tuttavia, questa versione richiede una discretizzazione spaziale molto più raffinata e, per questa ragione, non viene sempre settato nelle simulazioni dove le disponibilità di tempo ed hardware sono limitate.

## 5.7 Le altre Dictionaries utilizzate

Per lo svolgimento della simulazione, oltre ai documenti del codice principi già descritte vengono anche utilizzate le seguenti dictionaries:

- *triSurfaceExtract*;
- *refineMeshDict*;
- *topoSetDict*;
- *setFieldsDict*;
- *decomposeParDict* e *reconstructParDict*.

### 5.7.1 surfaceFeatureExtractDict

L'applicazione viene utilizzata per preparare le geometrie da aggiungere al dominio. Ogni CAD da aggiungere alla geometria viene posizionato nella cartella *constant/triSurface* in formato .STL. L'applicazione in oggetto prende in input il file .STL e ne estrae le superfici di contorno che scrive in diversi files contenuti nella stessa cartella *constant/triSurface*.

### 5.7.2 topoSetDict

Questa utility serve ad avere un controllo sulle entità della griglia di calcolo, punti, celle, facce. L'utilizzo finale può avere molte applicazioni diverse ma l'obiettivo è quello di selezionare ed isolare particolari zone di griglia per assegnare loro un nome e manipolarle, successivamente, in maniera diversa dalle restanti entità del dominio. Per esempio per assegnarle ad un raffinamento della griglia.

### 5.7.3 refineMeshDict

Essenziale durante la generazione della griglia, attraverso questa dictionary è possibile applicare dei raffinamenti a delle zone particolari di dominio selezionate, e distinte in precedenza, dall'applicazione *topoSetDict*.

Il raffinamento viene applicato nelle tre direzioni caratteristiche,  $x, y, z$ , dimezzando la dimensione della cella nella direzione inserita in input.

### 5.7.4 setFieldsDict

L'assegnazione dei valori di campo alle celle ed alle superfici, prima dell'applicazione dell'algoritmo di soluzione delle equazioni, in breve, il processo di inizializzazione, avviene grazie alla dictionary *setFieldsDict*. Le celle o le facce che vengono inizializzate sono precedentemente selezionate dall'applicazione *topoSetDict*.

### 5.7.5 decomposeParDict e reconstructParDict

Queste applicazioni permettono la prima di suddividere il dominio in gruppi di celle per il calcolo in parallelo. La scelta per la suddivisione del dominio può essere descritta come input durante la compilazione della dictionary posizionata nella cartella *system*. La seconda permette di ricostruire i sottodomini precedentemente decomposti per il calcolo parallelo per unire i risultati del calcolo in un'unica soluzione.

## 5.8 Gestione del codice e lancio della simulazione

La simulazione è strutturata in due cartelle, una rappresenta la cartella per la generazione della mesh dell'overset ed una contiene i dati per la mesh di *background* e del resto della simulazione. Lo schema logico consiste nel generare prima i due componenti della mesh in cartelle separate e, successivamente, unirle nella cartella di *background* che servirà a contenere tutti i documenti della simulazione globale.

Il flusso di passaggi che contraddistinguono lo svolgimento vero e proprio delle operazioni è contenuto nello script *Allrun.pre* ed *Allrun*. Questi fogli contengono le operazioni, una dopo l'altra, che si vogliono svolgere per avviare il calcolo della simulazione richiamando gli algoritmi che utilizzano i fogli di calcolo visti prima.

Nell'ordine, i passaggi svolti sono:

1. **Cartella Hull\_overset** È la cartella che contiene i documenti per la generazione della mesh di overset, il foglio con i passaggi da eseguire è il foglio *Allrun.pre* che contiene i seguenti passaggi:
  - (a) **triSurfaceExtract** La dictionary estrae i files .STL delle geometrie presenti nelal cartella *constant/triSurface*;
  - (b) **blockMeshDict** Esegue il comando per generare il box che contiene lo scafo, le cui superfici di contorno rappresentano il contorno dell'overset;
  - (c) **decomposeParDict**, si richiama la parallelizzazione per la generazione della mesh;
  - (d) **snappyHexMeshDict**. Durante questo passaggio le geometrie dello scafo vengono inserite nel blocco generato nel passaggio precedente tramite le regole descritte nel capito relativo alla dictionary *snappyHexMesh*;
  - (e) **reconstructParDict** Alla fine del processo di meshatura, il codice ricostruisce i dati nelle cartelle per generare un dominio ricostruito rispetto alla parallelizzazione.
  
2. **Cartella Background** Dopo aver generato la griglia di calcolo dell'overset, i passaggi si spostano nella cartella *background* per svolgere i seguenti passaggi:
  - (a) **blockMeshDict** Genera il parallelepipedo che assume la funzione descritta come *tank*. Il *tank*, in questo caso, rappresenta l'ambiente marino nel quale funziona la barca in esame;
  - (b) **topoSetDict** e **refineMeshDict** Per eseguire i rifimenti della griglia di calcolo sia per l'onda che per le zone vicine al blocco di overset;
  - (c) **topoSetDict** La selezione delle celle da assegnare al dominio esterno o all'overset avviene attraverso la *topoSetDict* dedicata;
  - (d) **setFields** Si inizializzano sia i campi di moto sia le zone del dominio;
  - (e) **resotre0Dir** Viene generata la cartella dell'istante iniziale, l'istante 0, dalla cartella delle condizioni al contorno con il nome *0.orig*;
  - (f) **decomposeParDict** Il dominio inizializzato viene decomposto per il calcolo in parallelo;
  - (g) **overInterDyMFoam** Viene lanciato il codice principale per il calcolo della fluidodinamica.

## 5.9 Unione degli effetti

L'effetto delle appendici e delle vele sulla dinamica della barca viene trasferita dalle simulazioni dedicate alla simulazione della carena tramite le forze ed i momenti sviluppati dalle appendici e dalle vele nelle stesse condizioni di posizione e navigazione della barca. Dato il movimento della barca intera, per un calcolo migliore e più realistico vengono corrette le simulazioni delle altre entità affinché la simulazione avvenga nelle condizioni di navigazione effettiva. La variazione delle forze è comunque molto limitata e non comporta una forte variazione della posizione del corpo ma aiuta ad aumentare la precisione nel calcolo. Il procedimento per arrivare a convergenza è il seguente:

1. Simulazione delle appendici con assetto neutro;
2. Simulazione della carena fino alla convergenza della posizione;
3. Correzione della posizione delle appendici e delle vele;
4. Correzione delle forze nel modello della carena e nuova condizione di convergenza della posizione.

La comunicazione tra le simulazioni avviene attraverso il passaggio dei momenti e delle forze agenti sulle appendici e sulle vele nella simulazione della carena come descritto

Nel caso in cui si decide di impostare una simulazione unica, la discretizzazione completa vede la somma del numero di celle delle simulazioni, un numero molto alto. Il vincolo di non poter dividere un'unica simulazione su diversi calcolatori provoca la lievitazione del tempo necessario per portare a termine il calcolo in tempi utili con un computer casalingo.

Oltre al numero di celle ciò che rallenta il calcolo è anche la dimensione delle celle più piccole delle appendici; dato il necessario affinamento in corrispondenza del bordo d'attacco dei profili ed il conseguente abbassamento del time-step per mantenere la condizione del numero di Courant.

La divisione delle componenti della barca e la confluenza degli effetti nella simulazione della sola carena porta ad una riduzione notevole del costo computazionale data la possibilità di lanciare le diverse simulazioni in contemporanea, mantenendo comunque tutti gli effetti presenti durante la navigazione.

Le simulazioni del caso studio vengono simulate con un singolo ciclo di correzione per la carena e le appendici mentre per le vele le forze vengono corrette analiticamente con le leggi trigonometriche. Nell'ingegneria navale che

coinvolge la progettazione completa delle barche a vela, lo sviluppo delle vele viene eseguito da terzi che si dedicano solo a questa mansione. Interfacciare due o più aziende rappresenta un rallentamento nelle procedure di progetto, per questo motivo, la manipolazione delle forze nel modello costruito viene svolta in maniera automatica con i dati di partenza provenienti dalla casa costruttrice delle vele.

## 5.10 Modifica delle simulazioni per il regime di navigazione simmetrico

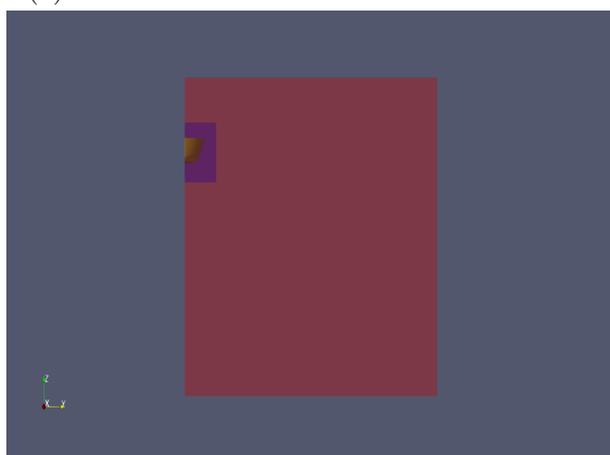
Una variante semplificativa della simulazione descritta sopra dove si studiano i tre gradi di movimento della carena, nasce dall necessità di analizzare la curva della resistenza della carena. Questo aspetto richiede lo sviluppo di simulazioni CFD che tengano conto di un caso in cui l'asse longitudinale della barca è allineata al vettore velocità ed i gradi di libertà presi in considerazione sono soltanto quelli di traslazione verticale e assetto di trim assunto dalla carena. Il risultato su cui si focalizza l'attenzione è la resistenza all'avanzamento sviluppata dalla carena.

Lo sviluppo di una simulazione a partire dal caso studio completo prevede la divisione del dominio sull'asse di simmetria della barca quindi il dimezzamento del dominio di partenza. L'ulteriore variazione prevede l'assegnazione della boundary condition sul piano che attraversa la simmetria del problema che, appunto, verrà settata come condizione di *symmetry*.

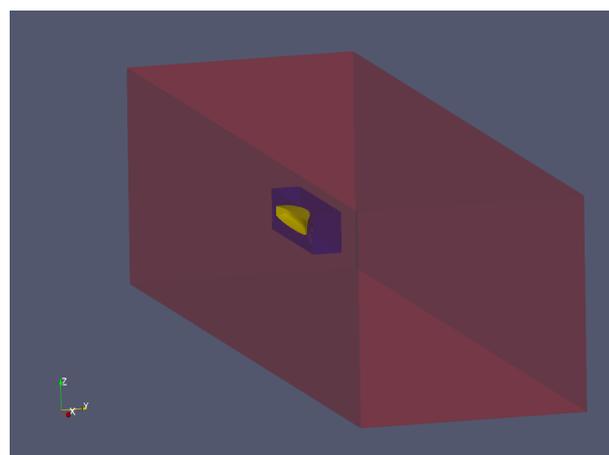
La discretizzazione spaziale del dominio risultante, le condizioni sulla dinamica e sul time-step restano invariati rispetto al caso completo. Il dominio delle simulazioni simmetriche, con due gradi di libertà, è il seguente:



(a) Vista laterale del dominio



(b) Vista frontale del dominio



(c) Vista del dominio in prospettiva

Figure 5.6: Dominio del codice di calcolo

La modifica si effettua per motivi di costo computazionale, infatti, lo studio del caso simmetrico si può effettuare anche mantenendo l'impostazione della simulazione per l'analisi del regime di navigazione su 3DoF con i dovuti adattamenti. Dalla divisione del dominio e dimezzando il numero di celle, si ottiene un dimezzamento dei tempi di calcolo a parità delle altre condizioni per questo è ottimale svolgere questa modifica.

## 5.11 Hardware disponibile per lo svolgimento delle simulazioni

La potenza di calcolo disponibile per svolgere le simulazioni della carena si basa su un computer di tipo gaming assemblato con i seguenti componenti principali:

- Processore Intel Core i7-8700 6 Core con hyper threading dissipato a liquido;
- Memoria RAM 16 GB DDR4-2666;
- Scheda Video nVidia GEFORCE GTX 1060;
- Archiviazione SSD 128 GB + HD 1 T.

Il sistema operativo installato sulla macchina è Windows 10 Pro e l'utilizzo di OpenFoam avviene tramite l'applicazione UBUNTU terminal, che permette di creare un ambiente di terminale proprio del sistema ubuntu direttamente nell'ambiente di Windows.

Le caratteristiche elencate descrivono nel dettaglio la strumentazione utilizzata per completare il calcolo implementato per quanto riguarda le simulazioni della carena. La descrizione della componentistica utilizzata e la comprensione della potenza di calcolo è indispensabile per comprendere il livello di precisione delle simulazioni svolte.

Per la soluzione di un dominio complesso, infatti, come è il dominio del caso in esame, possono essere necessarie più decine di cores e una ram con memoria tra i 32 GB ed i 64 GB. Il lievitare dei costi dell'hardware, però, pone un vincolo alla fattibilità del progetto.

La conoscenza dell'hardware disponibile offre la capacità di creare delle simulazioni funzionanti ma con le dovute limitazioni di costo, e quindi di precisione.

Le limitazioni che vengono attuate fanno riferimento alla discretizzazione spaziale. La fase critica della mesh, per le simulazioni del settore navale, è rappresentata dalla discretizzazione della *free surface*, ossia per la superficie di contatto tra aria e acqua, che dev'essere, per l'intero dominio, discretizzata molto finemente per descrivere in maniera corretta l'onda generata dalla barca stessa. Limitare la precisione in questo ambito vuol dire diminuire la

precisione di calcolo dell'onda. Una simulazione "alleggerita" però, non necessariamente è inutilizzabile per i fini progettuali anzi, è preferibile sviluppare modelli veloci che concentrino le esigenze di potenza di calcolo laddove è indispensabile un calcolo preciso, nell'ambito dei risultati, da applicare al progetto finale. Nella pratica dell'applicazione della computational fluid dynamics al progetto reale quindi, si trova il perfetto compromesso tra la precisione della simulazione richiesta e la potenza di calcolo necessaria.

Il compromesso trovato si adatta ad essere sviluppato per mezzo dell'hardware descritto sopra, una configurazione, tra l'altro, relativamente poco costosa e con componenti facilmente reperibili sul mercato.

# Chapter 6

## Analisi dei risultati

*L'ultimo passaggio di ogni simulazione è la manipolazione e l'analisi dei risultati. In letteratura questa operazione viene chiamata fase di post-processing.*

*Dalle equazioni di Navier-Stokes si ottengono le componenti di velocità, la pressione ed eventualmente la temperatura se vengono utilizzati dei modelli termodinamici. Durante la fase di post-processing, vengono utilizzate per il calcolo di grandezze come le forze applicate al corpo immerso nel fluido oppure i flussi termici sulla superficie del corpo.*

*Inoltre, il post-processing e l'analisi dei dati della simulazione, sono le operazioni attraverso le quali si analizzano i valori relativi al codice di calcolo ed alla numerica della simulazione, tramite i quali è possibile comprendere il livello e la correttezza della convergenza della simulazione.*

*Il capitolo è dedicato all'estrapolazione ed alla descrizione dei risultati delle simulazioni svolte ed all'analisi, nell'ottica dell'analisi prestazionale della barca a vela, dei dati in uscita dalle simulazioni fluidodinamiche.*

### 6.1 Convergenza ed analisi della simulazione

Le simulazioni costruite, hanno raggiunto la convergenza dei residui attraverso il settaggio sulla tolleranza impostato nel foglio *fvSolution*, ad ogni iterata, quindi la convergenza della soluzione delle equazioni all'interno dell'algoritmo pimple è garantita da questo vincolo.

Per quanto riguarda le forze agenti e gli spostamenti effettuati dalla carena non ci sono impostazioni automatiche all'interno della simulazione ma la con-

vergenza viene definita attraverso un algoritmo di post-processing scritto ad-hoc, che verifica che per un range di tempo fisico finale di  $0.5 \text{ s}$  l'oscillazione dei parametri non sia superiore allo  $0.5\%$  della media degli stessi.

Il tempo fisico risolto è di  $15\text{s}$ , tempo nel quale la barca segue il transitorio che la porta al punto di equilibrio ed assesta la posizione e tutte le forze ad essa applicate dal fluido. Considerando i settaggi descritti, l'hardware disponibile ed i criteri di convergenza si ottiene che il tempo di calcolo di una simulazione è di  $144$  ore, corrispondenti a  $6$  giorni (considerando che la parallelizzazione avviene su  $10$  dei  $12$  cores disponibili).

La validazione del modello in esame avviene tramite il confronto dei dati della fluidodinamica computazionale con quelli ottenuti attraverso il VPP utilizzato dalle aziende per lo sviluppo dei loro progetti fino ad ora.

## 6.2 Analisi sui dati numerici della simulazione

Nel seguito vengono descritti i dati risultanti dalle simulazioni inerenti alle caratteristiche numeriche del codice di soluzione utilizzato. I parametri che vengono osservati sono: il numero di Courant, il Wall  $y+$  e la Volume Fraction of water per valutare la ventilazione numerica.

### 6.2.1 Numero di Courant

Il numero di Courant è un parametro numerico critico per la stabilità e la convergenza della simulazione. È definito con la relazione:

$$Co = |a| \frac{\Delta t}{\Delta x} \quad (6.1)$$

Il numero di Courant, nel caso di simulazioni time-dependent, descrive il grado dell'avanzamento della soluzione nello spazio rispetto al passo temporale attraverso un parametro adimensionale.

È fondamentale mantenere il Courant al disotto dell'unità in linea teorica, per evitare che la soluzione "salti" delle celle nell'ambito di un *time-step*, e quindi generi instabilità del codice. Inoltre, in punti critici di ogni caso studio, è necessario mantenere il Courant entro un certo range.

I punti fondamentali per le simulazioni impostate sono la superficie di contatto aria/acqua e la zona di sovrapposizione tra la mesh di overset e la

mesh di *background*. In queste zone il numero di Courant dev'essere al disotto di un valore preciso minore dell'unità. Le celle che contengono la superficie di contatto tra aria ed acqua devono essere tali da garantire un numero di Courant minimo, intorno ad un valore di  $Co = 0.3$ .

Anche per le zone di interfaccia tra le due mesh, il metodo di interpolazione *inverse distance*, per un calcolo ottimizzato, richiede un Courant al disotto di  $Co = 0.5$ . Tenute in considerazione le caratteristiche riportate sopra, è necessario settare i parametri della simulazione: dimensione della mesh, time step e velocità, in maniera tale da ottenere il massimo *time-step* possibile che rispetti tutte le condizioni imposte dal numero di Courant, per aumentare la velocità di calcolo.

Durante l'impostazione del codice di calcolo è possibile impostare la simulazione con *time-step* variabile assegnando il numero di Courant massimo sia per le celle del dominio globale sia per le sole celle contenenti la superficie di contatto tra le fasi, nelle quali viene risolta l'equazione della Volume Fraction.

Nonostante il controllo sul numero di Courant massimo, è necessario osservare graficamente in valore calcolato nelle zone critiche.

Attraverso la dictionary *controlDict*, si richiede al codice principale di calcolare il numero di Courant in modo tale da poterlo analizzare graficamente nel software di post-processing.

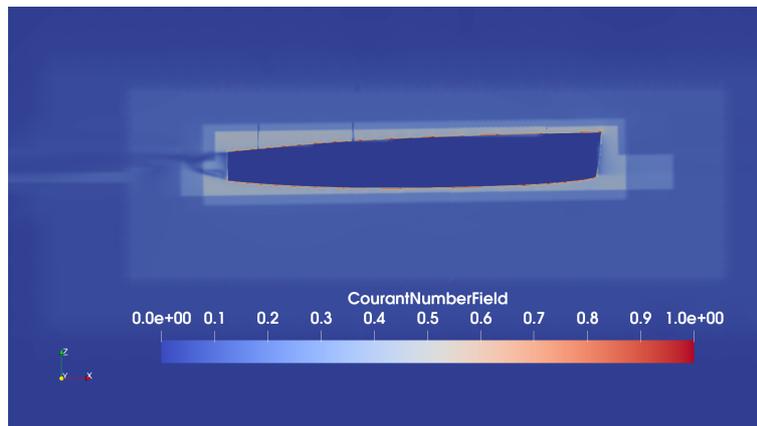


Figure 6.1: Distribuzione numero di Courant ottenuto nelle simulazioni

In figura 6.1 è riportata la scena che descrive la distribuzione del numero di Courant per le simulazioni descritte nel capitolo precedente. La scena è utile a vedere se le condizioni di vincolo imposte da tale parametro sono rispettate

e dove, eventualmente sono necessarie delle modifiche, o nella mesh o nella discretizzazione temporale.

Con i settaggi descritti si ottiene una distribuzione del Courant minore dell'unità per la maggior parte del dominio, l'unica eccezione vale per le celle di parete che, essendo molto raffinate richiedono un *time-step* che comprometterebbe la fattibilità del progetto per costo computazionale. Il fatto che le zone limite siano ristrette si può affermare che questo fenomeno non pregiudica l'utilità della simulazione. Il secondo vincolo sul valore del Courant viene imposto all'interfaccia tra *overset* e *background*. In queste zone si vede chiaramente dalla scena che il criterio di  $Co < 0.5$  è ampiamente rispettato.

### 6.2.2 Ventilazione numerica

La ventilazione numerica è un fenomeno di carattere numerico che si può verificare nelle simulazioni multi fase. La ventilazione numerica emerge dalla soluzione numerica ed approssimata dell'equazione della *volume fraction* e consiste in una scorretta valutazione di quest'ultima grandezza.

Durante gli studi prestazionali delle carene, questo fenomeno si verifica in particolare al disotto della carena, dove la superficie risulta, nella realtà, completamente "bagnata" ma per la ventilazione numerica la volume fraction di acqua risultante in alcune porzioni è  $\gamma_{water} < 1$ . Questo causa una scorretta valutazione della densità e della viscosità presenti in quell zona e, se eccessivamente estese, può portare ad una sottovalutazione delle forze agenti e quindi anche ad un calcolo non corretto dell'assetto assunto dalla barca. In alcuni casi, in particolare per i regimi plananti, per la configurazione della mesh la ventilazione numerica è eccessiva e va limitata aumentando il raffinamento nella zona di interfaccia tra le due fasi e limitando il *time-step* per diminuire il numero di Courant.

Una terza soluzione, più costosa a livello computazionale, è il raffinamento del prism layer. Svolgere questo passaggio contribuisce in maniera sostanziale alla limitazione della ventilazione numerica ma costringe al settaggio di un metodo di calcolo della turbolenza basato su algoritmi più complessi che richiedono una mesh maggiormente raffinata anche fuori dal prism-layer. A causa della mancanza di risorse hardware disponibili questa soluzione non è stata presa in considerazione.

In alcuni casi la ventilazione presente è accettabile e non compromette eccessivamente i dati calcolati e non viene ulteriormente corretta a vantaggio del costo computazionale. La seguente immagine riporta la frazione in volume di acqua presente al disotto della carena, corrispondente alla porzione immersa. Viene riportata la simulazione in cui la condizione di navigazione è più critica dal punto di vista della ventilazione numerica.

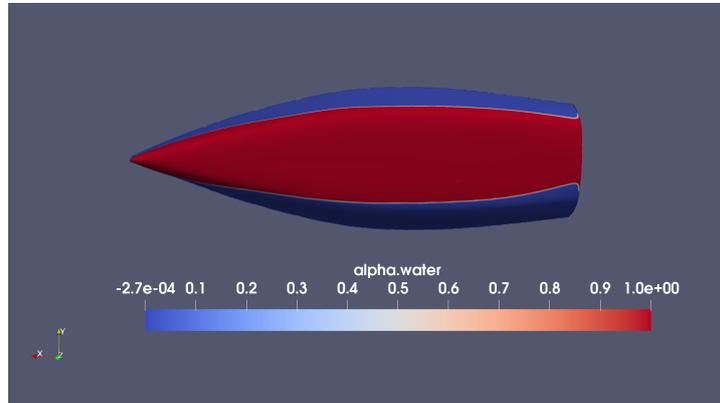


Figure 6.2: Distribuzione numero di Courant ottenuto nelle simulazioni

Le simulazioni svolte non hanno riscontrato criticità dal punto di vista della ventilazione numerica sia per la velocità limitata dell'acqua sia per il giusto compromesso di raffinatezza della mesh e del time step. Come si può osservare dalla figura 6.2 la *water volume fraction* della superficie bagnata della barca è ampiamente pari a 1, a confermare che, numericamente, la superficie della carena si affaccia ad un fluido con caratteristiche esattamente pari a quelle dell'acqua.

### 6.2.3 Wall $y^+$

Parametro numerico necessario al corretto funzionamento del modello per il calcolo della turbolenza è il *wall  $y^+$* . Il codice di calcolo utilizzato, il  $\kappa - \epsilon$  necessita di un valore di  $y^+$  che sia maggiore di  $y^+ > 30$ . Fisicamente, questo aspetto, richiede che il buffer layer dello strato limite sia interamente contenuto nella prima cella affacciata alla parete.

Alla fine della simulazione, durante la valutazione dei parametri numerici, si controlla che anche il parametro in oggetto rispetti le condizioni richieste. Nel caso particolare delle simulazioni svolte, la zona di maggiore criticità per questo parametro è la superficie di carena immersa in acqua. Per le simulazioni ottenute si ottiene la seguente distribuzione della grandezza *wall  $y^+$* .

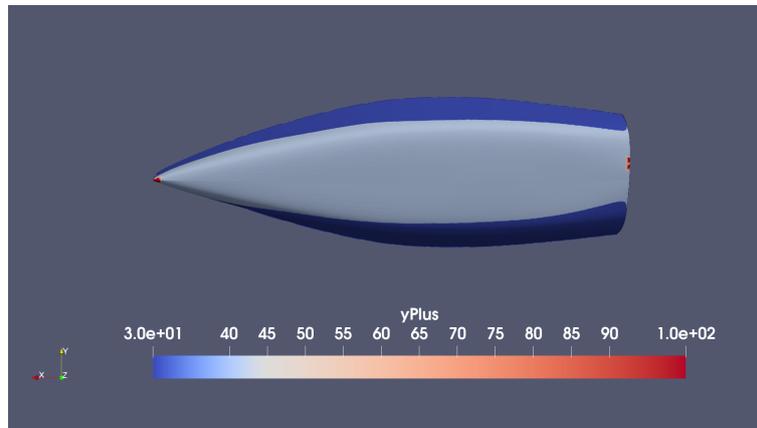


Figure 6.3: Distribuzione numero di Courant ottenuto nelle simulazioni

Anche le condizioni per il corretto funzionamento del modello di turbolenza sono rispettate. Dall'immagine 6.3 si può confermare che il valore di  $y+$  rispetta la condizione imposta dal modello per la turbolenza  $k-\epsilon$ . Esistono alcune zone in cui il valore del parametro in esame è fortemente differente dalla media calcolata, questo può essere dato da alcune singole celle di mesh che hanno una qualità bassa oppure dal fatto che in alcune zone, come a prua e a poppa possono esserci zone di distacco della vena fluida o zone di ricircolo. Questo aspetto non porta a problematiche rilevanti, perciò viene accettato durante la fase di *post-processing*.

### 6.3 Analisi dei parametri fisici ottenuti dalle simulazioni

Dopo aver constatato che i dati sulla numerica del codice siano idonei ad una corretta convergenza delle simulazioni, viene svolta la fase di post-processing più importante, la valutazione dei parametri fisici ottenuti mediante le simulazioni.

I parametri di funzionamento corrispondono alle caratteristiche del regime di navigazione e sono i risultati per i quali le simulazioni vengono costruite. Queste grandezze verranno poi utilizzate nel progetto della barca finale. L'analisi inizia con la descrizione della curva di resistenza della carena mediante le simulazioni semplificate della sola carena. Successivamente, si svolgono le valutazioni sul comportamento globale della barca nei tre regimi di navigazioni fondamentali testati.

### 6.3.1 Caratterizzazione della curva di resistenza della carena

L'analisi preliminare della carena si effettua tramite la realizzazione della curva della resistenza. Questa, come visto nei capitoli precedenti, descrive l'andamento della resistenza al variare della velocità di avanzamento della barca. Lo studio viene effettuato con le simulazioni CFD semplificate, in cui il campo di moto è simmetrico e si considerano due soli gradi di libertà, anch'essi simmetrici, della carena: la traslazione verticale e l'angolo di trim.

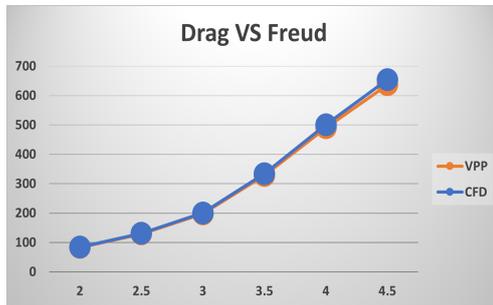
Le simulazioni svolte hanno portato ai seguenti risultati:

$V$ [m/s]	CFD			VPP		
	$h_z$ [m]	$\theta_y$ [deg]	$F_x$ [N]	$h_z$ [m]	$\theta_y$ [deg]	$F_x$ [N]
2	-0.00317	-0.359	84.20	-0.0031	-0.35	83.26
2.5	-0.00991	-0.429	130.90	-0.0098	-0.42	129.12
3	-0.0171	-0.613	200.42	-0.017	-0.60	197.25
3.5	-0.0247	-1.243	334.26	-0.024	-1.22	329.28
4	-0.0245	-2.087	500.86	-0.024	-2.04	493.14
4.5	-0.0192	-2.786	654.32	-0.019	-2.71	637.24

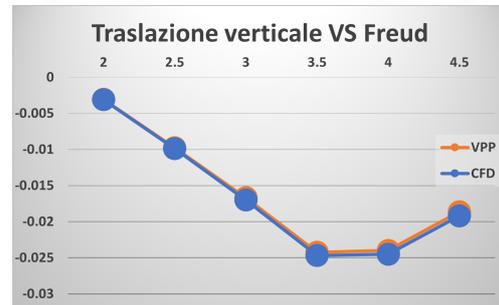
Table 6.1: Risultati numerici delle simulazioni in regime simmetrico

$V$ [m/s]	Errore Assoluto			Errore Relativo		
	$h_z$ [m]	$\theta_y$ [deg]	$F_x$ [N]	$h_z$ [m]	$\theta_y$ [deg]	$F_x$ [N]
2	$-3.41e^{-5}$	-0.0028	0.94	1.12	0.97	1.12
2.5	$-12.47e^{-5}$	-0.0057	1.78	1.36	1.26	1.36
3	$-31.45e^{-5}$	0.0095	3.17	1.58	1.85	1.58
3.5	$-45.94e^{-5}$	0.025	5.98	1.79	1.86	1.79
4	$-52.97e^{-5}$	-0.049	10.72	2.14	2.15	2.14
4.5	$-57.98e^{-5}$	-0.080	17.08	2.61	3.02	2.61

Table 6.2: Errore dei risultati numerici delle simulazioni in regime simmetrico



(a) Resistenza al variare del Freud per il caso simmetrico



(b) Traslazione verticale al variare del Freud per il caso simmetrico



(c) Angolo di trim al variare del Freud per il caso simmetrico

Figure 6.4: Analisi del regime simmetrico calcolato dalle simulazioni CFD

Il funzionamento in condizioni ottimali del Melges 24 consiste nella navigazione in regime planante. Infatti, come descritto nel capitolo dedicato, è una barca molto leggera disegnata con geometrie che favoriscono questo tipo di navigazione.

Dall'analisi tramite la Computational Fluid Dynamics, si possono osservare numerosi dati che descrivono il comportamento della carena posta in navigazione in regime simmetrico a diverse velocità dell'acqua. A basse velocità, minori di  $V = 3m/s$  il regime di navigazione è ancora fortemente dislocante, si può vedere in figura 6.5 come il sistema di onde generato dalla barca è un sistema formato dalle onde divergenti e dalle onde trasversali, non solo, più lunghezze d'onda attraversano la lunghezza della carena esattamente come avviene nel regime dislocante.

Un altro aspetto importante da considerare è l'angolo di Trim che assume la carena. Nel regime planante, infatti, questa grandezza assume valori notevoli. Osservando il grafico in figura 6.4c, si può osservare come il trim sia molto limitato nelle basse velocità,  $V \simeq 3m/s$ . Aumentando ulteriormente la

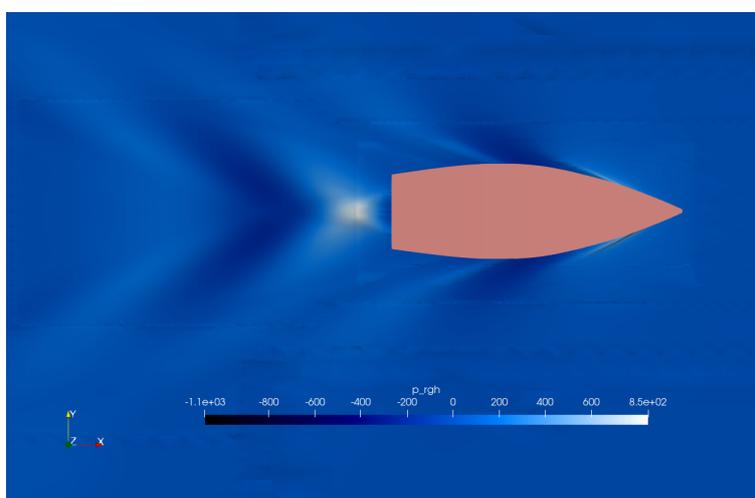


Figure 6.5: Scena sistema d'onda in regime dislocante con  $V=3$  m/s

velocità, passando a  $V = 4$  m/s, si osserva una rapidissima impennata della barca a raggiungere valori di trim decisamente più consistenti.

Unendo la precedente constatazione all'osservazione dell'onda generata dalla barca in questo regime si può confermare che ci si trova al passaggio di regime da dislocante a planante. Il numero di creste (colorate di azzurro più chiaro) che attraversano la lunghezza della carena corrisponde ad una lunghezza d'onda, come descrive la teoria .

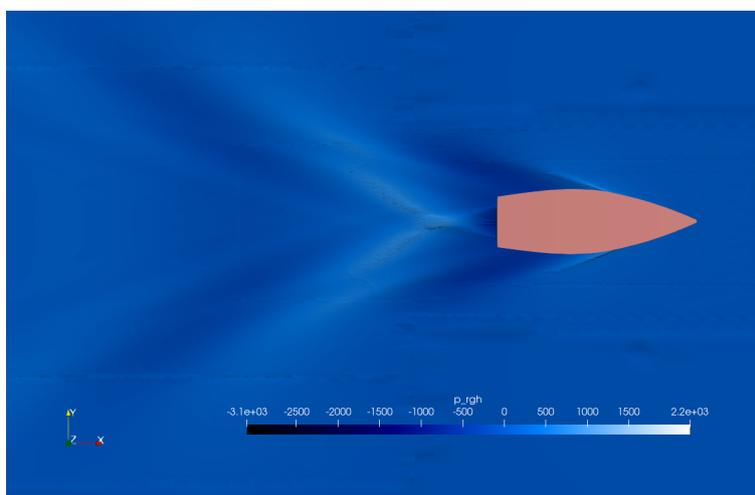


Figure 6.6: Scena sistema d'onda in regime planante con  $V=4.5$  m/s

L'evoluzione dell'onda al variare delle velocità testate sono riportate in figura:

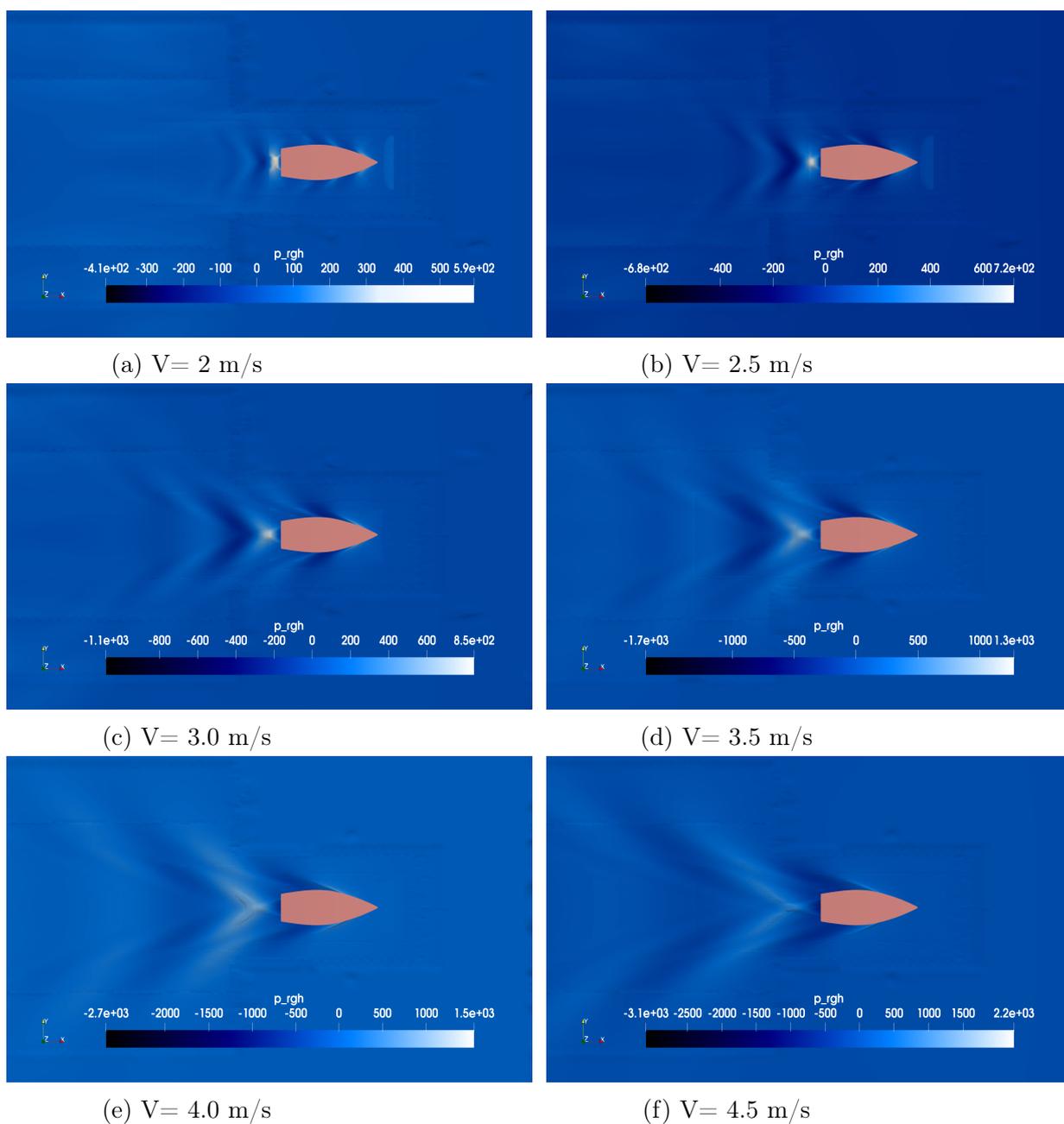


Figure 6.7: Evoluzione del sistema d'onde generato dalla barca

Osservando le scene realizzate durante il post processing delle simulazioni contenute in figura 6.7 si nota come, lo sviluppo dell'onda qualche metro dietro la barca sia differente in funzione della velocità. La vista dall'alto del sistema di onde generate descrive perfettamente il regime di navigazione. Nella

figura 6.7a si vede chiaramente che le creste dell'onda entro la lunghezza della carena sono superiori all'unità; le zone chiare, a rappresentare i punti più alti dell'onda, sono tre. Ad una velocità di  $2 m/s$  quindi è dimostrato il regime altamente dislocante. Osservando l'estremo opposto, la simulazione alla velocità di  $V = 4.5 m/s$ , in figura 6.7f, si vede che l'onda generata è molto più allungata e solo un'unica cresta attraversa lo scafo della barca. In questo caso la condizione di navigazione è ampiamente una condizione planante.

Tenendo in considerazione l'aspetto fisico che contraddistingue il regime dislocante dal regime planante, la differenza dello sviluppo dell'onda deriva anche da aspetti numerici.

Mantenendo la stessa mesh ottimizzata per il regime a  $V = 3 m/s$  la discretizzazione verticale e nelle direzioni laterale e longitudinale non è abbastanza raffinata per un'onda che, alla velocità di  $V = 2 m/s$ , è più bassa e meno smooth di quella sviluppata a velocità più alte. Il risultato è che l'onda non viene calcolata perfettamente e risulta smorzata. Come si può vedere dalle scene, però, lo smorzamento inizia circa ad una lunghezza barca a partire dalla poppa, perciò, nell'ambito dello studio prestazionale e del calcolo delle forze agenti sulla barca non ha un effetto rilevante.

### 6.3.2 Analisi del regime di navigazione e del sistema di onde generato dalla barca

Terminata l'analisi delle simulazioni simmetriche si passa all'analisi delle simulazioni in andatura reale. Di queste si analizza la configurazione e l'assetto di navigazione assunto dalla barca ed è possibile osservare il sistema di onde generato dalla carena nelle diverse condizioni di navigazione per comprendere il regime di navigazione e analizzare l'entità della resistenza d'onda generata dalla carena. Come descritto sopra è utile capire se si naviga in regime planante.

Per una determinata velocità del vento, pari a  $w = 8 kn$  vengono analizzate tre andature caratteristiche: il traverso, la bolina e il gran lasco. Rispettivamente, l'angolo apparente del vento rispetto alle vele per le tre andature è:  $windangle = 90 deg$ ,  $windangle = 41 deg$  e  $windangle = 148 deg$ .

## Andatura di traverso

Input	Velocità [m/s] 3.645	Scarroccio [deg] 2.1	Avanzamento [m/s] 3.643	Scarroccio [m/s] 0.241
Appendici	Laterale [N] -1017.5	Longitudinale [N] 13.8	Sbandamento [Nm] -56.5	Trim [Nm] -1220.5
Vele	Laterale [N] 630.0	Longitudinale [N] 375.0	Sbandamento [Nm] -2313.5	Trim [Nm] 1350.2
Equipaggio	Longitudinale [m] 3.3	Trasversale [m] -0.90	Verticale [m] 0.6	Peso [N] 1471

Table 6.3: Velocità calcolate dal VPP in input alla simulazione per l'andatura di traverso

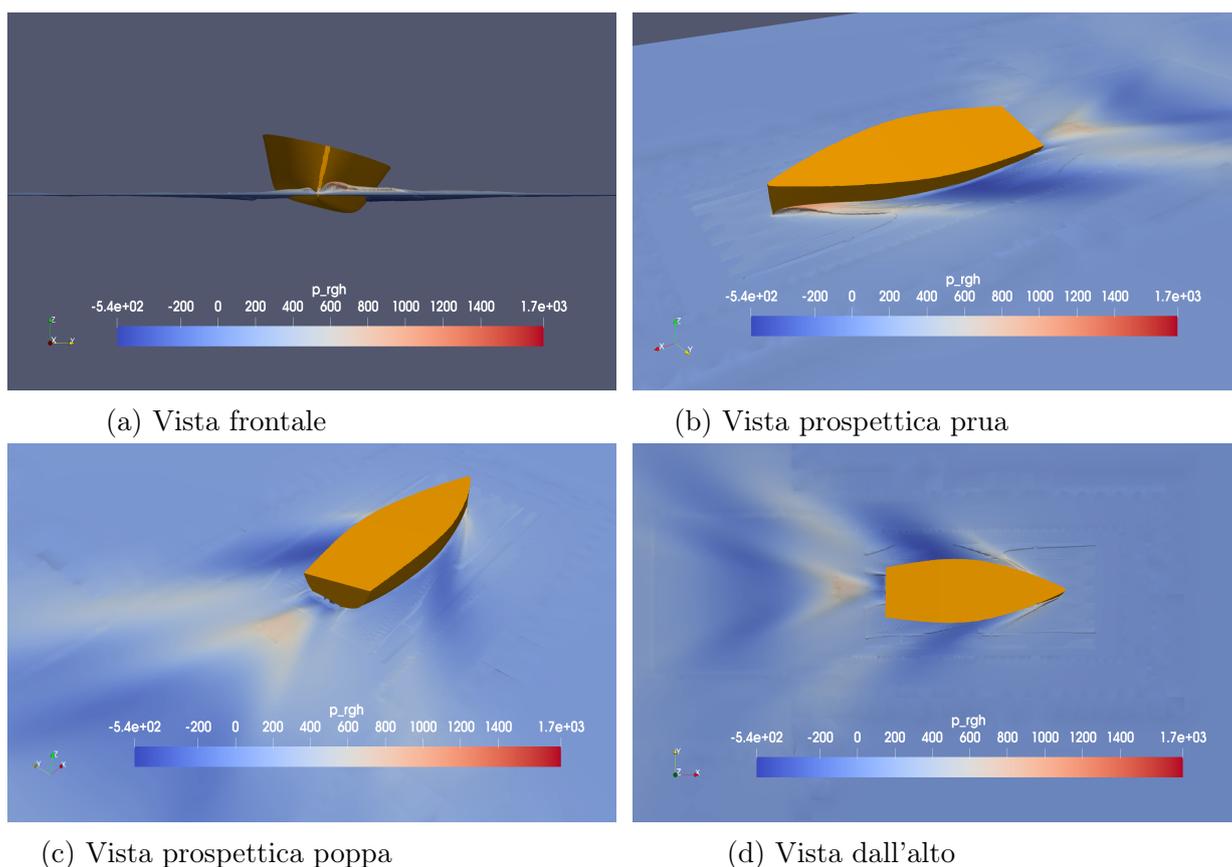


Figure 6.8: Risultati dell'andatura di traverso

**Andatura di bolina**

Input	Velocità [m/s] 2.643	Scarroccio [deg] 4.37	Avanzamento [m/s] 2.635	Scarroccio [m/s] 0.363
Appendici	Laterale [N] -980.8	Longitudinale [N] 15.8	Sbandamento [Nm] -1056.0	Trim [Nm] -58.5
Vele	Laterale [N] 657.2	Longitudinale [N] 380.0	Sbandamento [Nm] -2365.2	Trim [Nm] 1368.1
Equipaggio	Longitudinale [m] 3.3	Trasversale [m] -0.90	Verticale [m] 0.6	Peso [N] 1471

Table 6.4: Dati in input alla simulazione per l'andatura di bolina

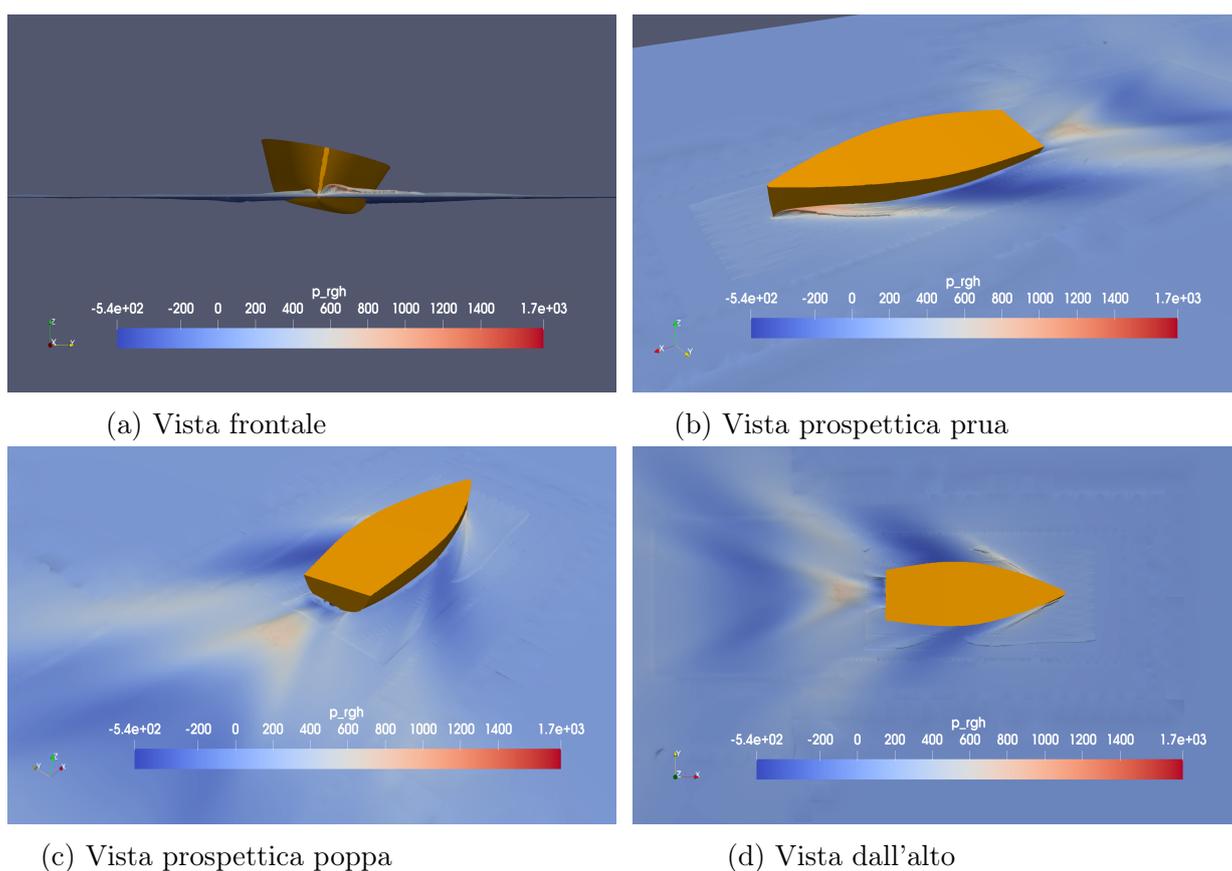


Figure 6.9: Risultati dell'andatura di bolina

## Andatura di gran lasco

Input	Velocità [m/s] 3.374	Scarroccio [deg] 0.52	Avanzamento [m/s] 3.374	Scarroccio [m/s] 0.055
Appendici	Laterale [N] -89.2	Longitudinale [N] -60.7	Sbandamento [Nm] -154.0	Trim [Nm] -95.8
Vele	Laterale [N] 176.6	Longitudinale [N] -313.1	Sbandamento [Nm] -635.7	Trim [Nm] 1127.1
Equipaggio	Longitudinale [m] 3.2	Trasversale [m] -0.15	Verticale [m] 0.6	Peso [N] 3090

Table 6.5: Dati in input alla simulazione per l'andatura di gran lasco

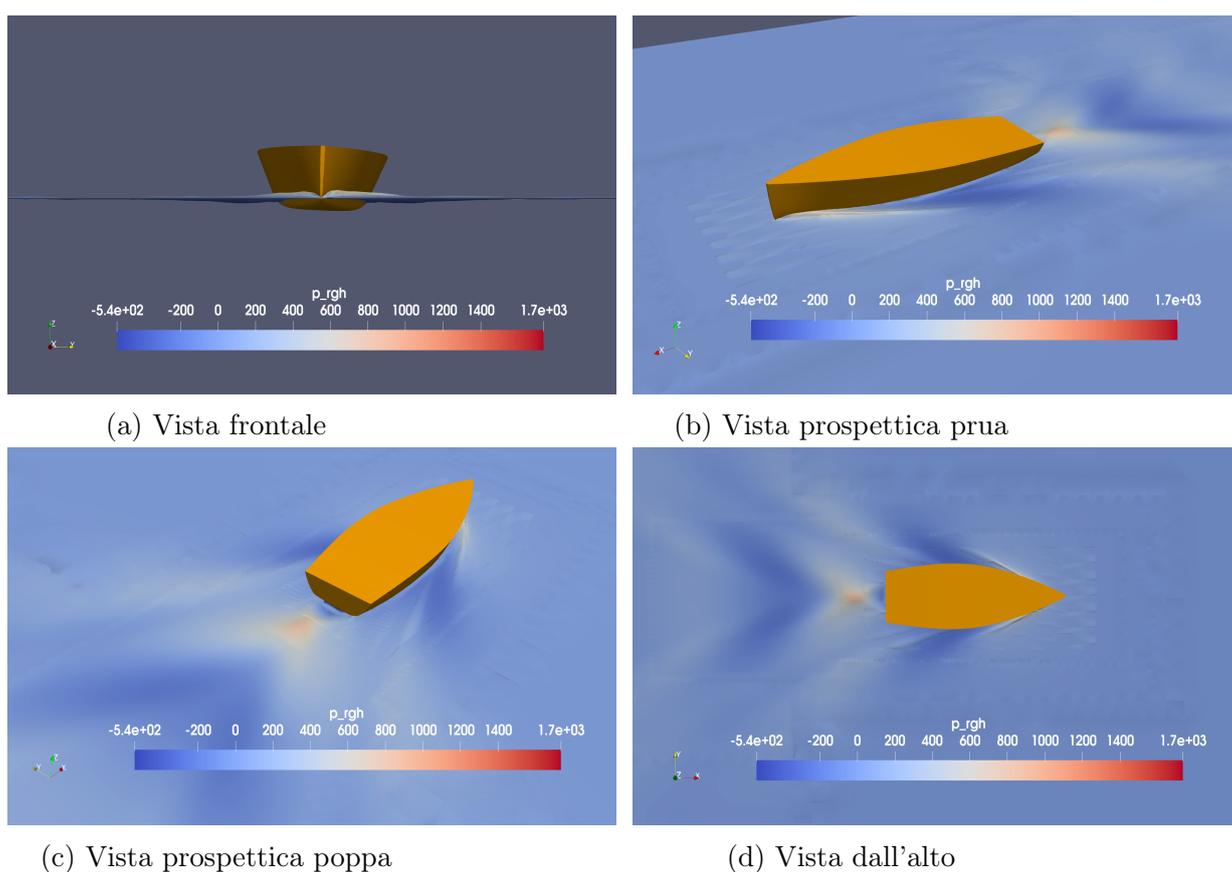


Figure 6.10: Risultati dell'andatura di gran lasco

### Dinamica della carena

Le grandezze ottenute mediante le analisi CFD circa la dinamica della barca riguardano l'assetto nelle componenti di:

- Traslazione verticale;
- Rotazione sull'asse longitudinale;
- Rotazione sull'asse trasversale;
- Forza longitudinale.

I valori ottenuti sono riportati in tabella per le diverse condizioni di navigazione:

Andatura	CFD				VPP			
	$h_z[m]$	$\theta_x[deg]$	$\theta_y[deg]$	$F_x[N]$	$h_z[m]$	$\theta_x[deg]$	$\theta_y[deg]$	$F_x[deg]$
Bolina	-0.0255	-10.897	-0.340	373.5	-0.0251	-10.490	-0.334	366.6
Traverso	-0.021	-12.615	-0.241	373.8	-0.0207	-12.17	-0.237	367.2
Gran lasco	-0.015	-2.293	-0.801	302.7	-0.0148	-2.26	-0.792	298.9

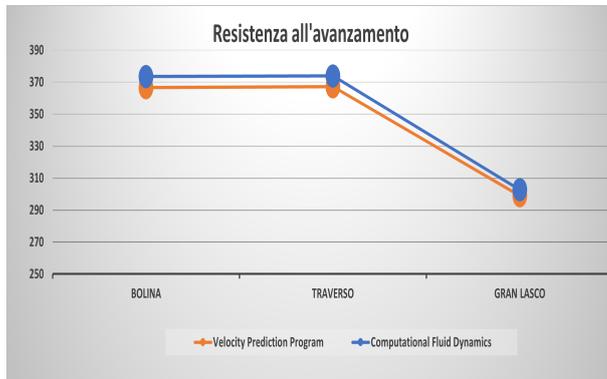
Table 6.6: Risultati numerici della barca completa

La differenza nel calcolo delle incognite tra VPP e CFD è riportata nella seguente tabella:

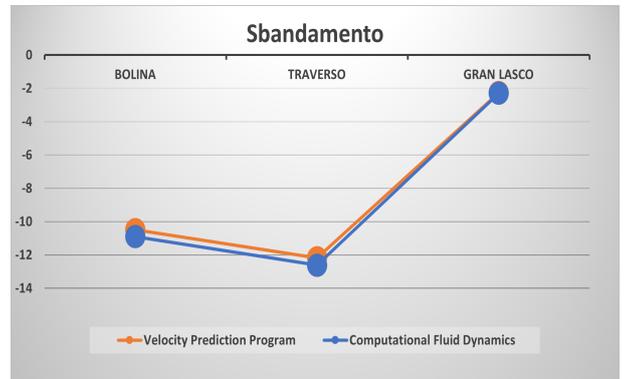
Andatura	Errore Assoluto				Errore Percentuale			
	$h_z[m]$	$\theta_x[deg]$	$\theta_y[deg]$	$F_x[N]$	$h_z[m]$	$\theta_x[deg]$	$\theta_y[deg]$	$F_x[deg]$
Bolina	$-0.40e^{-3}$	-0.41	-0.0060	6.94	1.56	3.76	1.75	1.86
Traverso	$-0.34e^{-3}$	-0.44	-0.0040	6.65	1.63	3.52	1.64	1.78
Gran lasco	$-0.18e^{-3}$	-0.038	-0.0092	3.72	1.21	1.65	1.15	1.23

Table 6.7: Errore numerico tra le simulazioni CFD e i modelli VPP

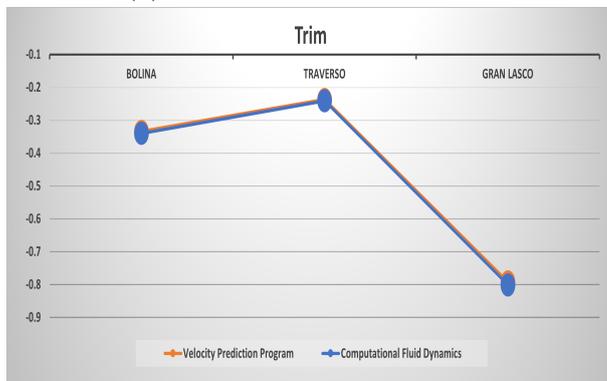
Nell'ambito della navigazione a vela l'andatura più veloce, a parità di vento apparente, è l'andatura di *traverso*. Durante questa andatura le vele sviluppano una notevole forza laterale che contribuisce a uno sbandamento notevole della barca. In questo caso l'equipaggio deve spostarsi sul lato alto della carena per sviluppare il momento antisbandamento utile a non fare



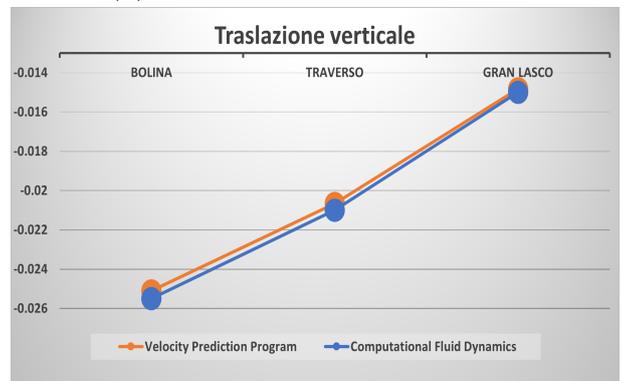
(a) Resistenza



(b) Sbandamento



(c) Trim



(d) Traslazione verticale

Figure 6.11: Andamento grafico dei risultati ottenuti mediante la CFD e il VPP

"scuffiare" (come si dice in gergo) la barca. Durante questa andatura la barca si trova in una posizione sulla verticale più alta e presenta uno sbandamento maggiore dell'andatura di bolina. L'angolo di trim è contenuto in tutte le andature, questo a significare che la posizione lungo l'asse longitudinale dell'equipaggio è una posizione che garantisce l'equilibrio delle forze rispetto al piano longitudinale. Questa posizione però, non favorisce la transizione a regime planante, che sarebbe facilitata da una posizione più appoppata dell'equipaggio. Con forti momenti sbandanti però, l'equipaggio stesso si posiziona nel punto più lontano dall'asse longitudinale in maniera di massimizzare il momento opposto a quello di vele e deriva. Il punto più lontano, tuttavia, si trova poco dietro il punto centrale sulla lunghezza della carena. Mentre per la bolina la situazione è molto simile all'andatura precedentemente descritta, le cose cambiano radicalmente per la terza andatura, il gran lasco. Durante questa andatura lo scarroccio è decisamente limitato e, di con-

seguenza, anche lo sbandamento. Non è necessario quindi che l'equipaggio si posizioni sul fianco della carena, in un punto più possibile lontano dall'asse di simmetria. Da qui la possibilità di creare uno sbilanciamento sull'asse di trim per cercare di favorire il regime planante. Come si osserva dalle immagini, nonostante la posizione dell'equipaggio più appoppata, non si raggiunge ancora completamente il regime planante.

Analizzando, invece, il confronto con i dati provenienti dal Velocity Prediction Program, si nota come l'errore sia decisamente limitato per tutte le grandezze in *output*. Questo aspetto garantisce la correttezza del codice di calcolo fluidodinamico utilizzato. Interessante risulta il fatto che nonostante l'utilizzo del principio di sovrapposizione degli effetti e nonostante l'utilizzo di una discretizzazione "leggera", non proibitiva, il calcolo risulta comunque affidabile e utilizzabile per il progetto di una barca di caratteristiche simili al MELGES 24.

### Distribuzione della pressione sulla carena

Essenziale durante il progetto della carena di una barca a vela è la comprensione della distribuzione delle pressioni sulla superficie della carena stesa. Le figure di seguito raffigurano il comportamento di questa grandezza per le condizioni di navigazione di bolina, traverso e gran lasco.

Le scene contenenti la distribuzione di pressione sono utili per diversi motivi. Nella progettazione strutturale della carena è essenziale comprendere come sono distribuiti gli sforzi durante le diverse condizioni di navigazione possibili. Negli studi più evoluti è possibile "tabellare" le distribuzioni di pressione disegnate in figura 6.12 per riportare le pressioni calcolate nelle simulazioni CFD in simulazioni strutturali basate sul calcolo agli elementi finiti rendendo disponibili modalità di progettazione complesse ma molto complete e orientate al miglioramento avanzato del prodotto finale.

Indipendentemente dalle necessità della progettazione strutturale della barca le scene di pressione sono utili a svolgere delle considerazioni sul funzionamento in generale. Osservando i dati si può stabilire nuovamente che le andature traverso e bolina siano simili tra di loro, si nota come il campo di pressione sia molto simile nella configurazione ma rappresenta pressioni più intense nell'andatura di traverso. La barca naviga con un angolo di sbandamento notevole (intorno ai  $11deg$ ) e per questo il campo di pressione è fortemente asimmetrico; in questi casi la carena è più sollecitata sul fianco che si immerge in acqua. Per queste andature, inoltre, si vede come la superficie immersa sia minore, aspetto che giustifica una maggiore componente di resistenza emergente dalle pressioni e una resistenza d'attrito viscoso più

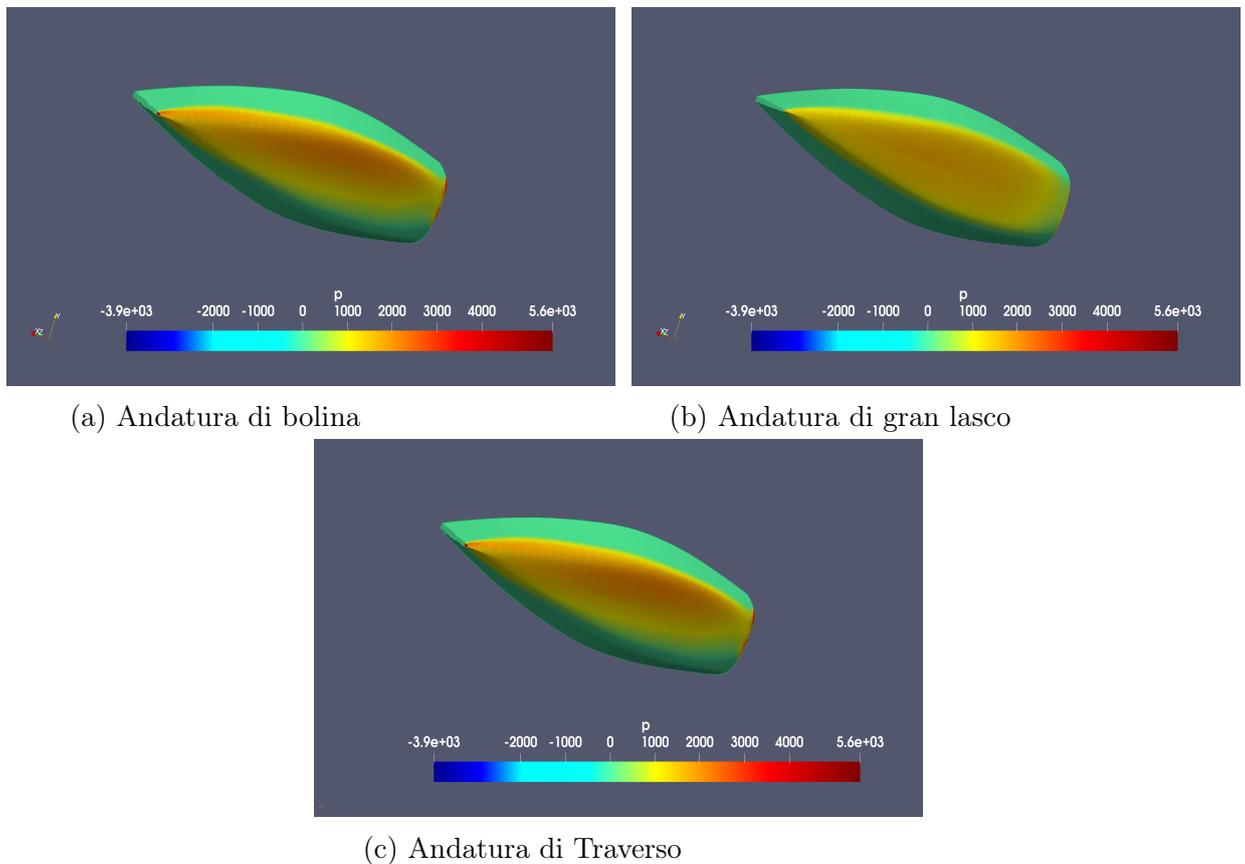


Figure 6.12: Distribuzione di pressione sulla carena in andatura

limitata.

L'andatura di gran lasco, invece, manifesta le differenze rispetto alle altre due andature principali anche dal punto di vista delle pressioni che il fluido esercita sulla carena. In questo caso il regime di navigazione è molto simile ad un regime simmetrico, perché lo scarroccio è basso e di conseguenza anche lo sbandamento è limitato. L'intensità della pressione è minore durante l'andatura di poppa. In questo caso la superficie immersa della carena è maggiore e quindi ci si aspetta che la resistenza complessiva sia composta da una parte maggiore dovuta agli sforzi viscosi rispetto alle altre due andature.

### Composizione della forza di resistenza

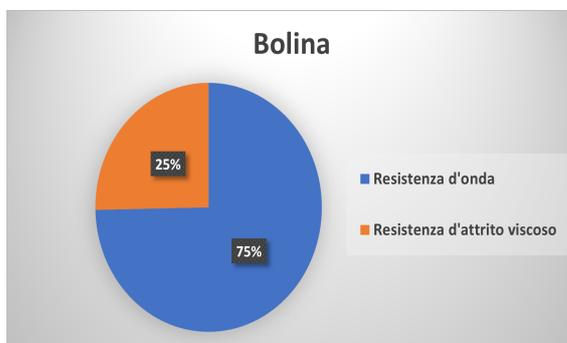
La resistenza, come descritto nel capitolo 2, ha più componenti. Parte della resistenza è data dagli sforzi d'attrito viscoso, parte dalla generazione dell'onda ed una piccola porzione è data dai gradienti di pressione lungo la superficie della barca. Durante il calcolo fluidodinamico, il codice è in grado di calcolare gli sforzi di taglio derivanti dalla viscosità del fluido; da questa considerazione, integrando gli sforzi sulla superficie della carena bagnata da un fluido o dall'altro giunge al calcolo della forza complessiva dovuta alla viscosità.

Derivano, invece, dal calcolo delle pressioni, le forze di resistenza d'onda e di resistenza per gradiente di pressione, quella componente che nel capitolo 2 viene chiamata *Resistenza di pressione* e rappresenta una minima parte della resistenza presente.

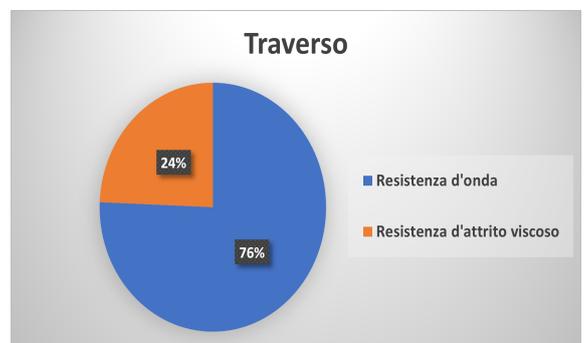
Nelle figure sottostanti sono contenuti i grafici che descrivono la composizione delle resistenze calcolate dalla simulazione CFD costruita. Dato che la resistenza d'onda è una componente prevalente rispetto alla resistenza di pressione e la carena del MELGES 24 presenta linee relativamente *smooth*, tutta la porzione di forza dovuta alla pressione viene attribuita alla resistenza d'onda.

Andatura	Resistenza d'onda [N]	Resistenza d'attrito viscoso [N]
Bolina	279.50	94.0
Traverso	282.91	90.89
Gran lasco	167.72	134.98

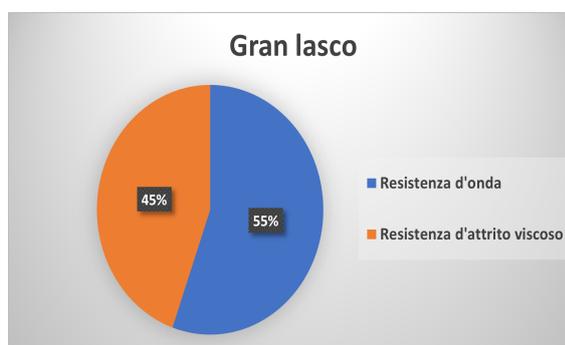
Table 6.8: Composizione della Resistenza all'avanzamento



(a) Andatura di bolina



(b) Andatura di traverso



(a) Andatura di gran lasco

Figure 6.14: Composizione della resistenza complessiva sviluppata dalla carena

La comprensione delle cause della resistenza della barca è utile per operare in maniera mirata nell'ambito dell'ottimizzazione prestazionale. Le andature di bolina e di traverso sono critiche dal punto di vista della resistenza d'onda dato il loro regime altamente asimmetrico. Sia lo sbandamento che lo scarroccio contribuiscono a favorire l'asimmetria della navigazione che porta il punto d'arresto del fluido a coincidere in una zona sul fianco bagnato della barca. Da questo punto, di massima pressione il flusso si sposta lungo i fianchi della carena e sarà costretto a forti variazioni di direzione vicino alla prua, motivazione del fatto che la resistenza dovuta alle pressioni è maggiore. L'onda di prua che si genera in prossimità del punto d'arresto è un'onda piuttosto alta, questo conferma il grande intervento della resistenza d'onda sulla resistenza totale. L'andatura di gran lasco, molto più simile al regime simmetrico, assicura che il punto d'arresto sia nei pressi della prua e quindi il flusso d'acqua è meno deviato, rispetto agli altri casi, proseguendo lungo la lunghezza della carena.

Dal punto di vista dell'analisi prestazionale, la soluzione al problema della minimizzazione della resistenza all'avanzamento è complicata ed il risultato finale è caratterizzato dal miglior compromesso. Per migliorare le andature di risalita del vento si può intervenire sulla forma della prua, aumentando la curvatura delle linee in maniera tale da smorzare le rapide variazioni di direzione del flusso. Questa operazione, tuttavia, peggiora le caratteristiche dell'andatura di poppa in quanto porta ad una geometria dall'area frontale maggiore, che sfavorisce l'andatura simmetrica.

Per migliorare, invece, le andature a favore di vento si dovrebbe operare sulla dimensione dell'area frontale. Variando la geometria in maniera concorde a

questa logica si genera una geometria più spigolosa che sfavorirebbe totalmente le andature di risalita del vento.

La geometria migliore, quindi, si può trovare con una logica iterativa, lavorando su simulazioni CFD come quelle descritte nel seguente lavoro costruite ad-hoc per le barche e le andature che l'imbarcazione sarà chiamata a sostenere. Sviluppando più modelli si possono mettere a confronto i dati numerici e le scene cromatiche di forze, velocità, pressioni, superficie bagnata ecc... per creare un confronto rigoroso che metta in luce anche piccole differenze altrimenti non facilmente osservabili.

## 6.4 Utilizzo dei dati nell'ambito del progetto

Il processo di sviluppo della simulazione CFD inerente al comportamento di una barca a vela durante il suo funzionamento in acqua, viene svolto per ottenere dei dati numerici utili ad un progetto. Tutti questi dati intervengono dopo i calcoli preliminari che portano al primo disegno della barca a vela. La prima configurazione del mezzo avviene grazie all'esperienza dei progettisti e a garanzia dei requisiti principali, per esempio se è necessario soffermarsi sul disegno di una carena dislocante o planante, oppure se la barca è una deriva o una barca a chiglia rigida. Questi primi requisiti di progetto dipendono dalle esigenze dell'armatore e dalle richieste dei clienti.

La fluidodinamica computazionale, quindi, interviene durante il perfezionamento del primo progetto, per la riduzione della resistenza all'avanzamento, per il miglioramento delle caratteristiche sulla dinamica della barca. La possibilità di svolgere gli studi descritti in maniera economica rispetto a test fisici, è una grande conquista per gli ingegneri che possono svolgere numerose prove in condizioni anche molto diverse tra di loro, senza dover ricorrere alla costruzione di complessi modelli o prototipi.

## Chapter 7

# Conclusioni e progetti futuri

Le simulazioni per l'analisi della navigazione della barca a vela di classe Melges 24 tramite la Computational Fluid Dynamics rappresentano uno strumento molto valido per la progettazione di barche a vela di dimensioni confrontabili alla suddetta classe velica. La possibilità di svolgere simulazioni fluidodinamiche che comprendono l'effetto di tutte le componenti della barca a vela permette di avere una visione realistica della barca in acqua senza ricorrere a costose prove in vasca che non si potrebbero effettuare per ogni nuovo progetto di categoria.

L'automazione dello studio è imposto dalla divisione in tre schemi delle diverse componenti: vele, appendici e carena, indispensabile per non appesantire in maniera accettabile i tempi di calcolo. La difficoltà durante la simulazione, infatti, proviene dalla schematizzazione delle forze agenti e dal trasporto degli effetti tra un dominio esterno al dominio della carena. Tuttavia, come descritto nell'elaborato, il settaggio corretto di tutti i parametri porta ad risultati coerenti con quelli ottenuti attraverso il VPP e coerenti anche con i dati osservati durante le prove sperimentali svolte in mare. Si può affermare, quindi, che le simulazioni costruite sono un metodo affidabile e veloce per effettuare lo studio globale del regime di navigazione in maniera esaustiva.

Il processo di validazione delle simulazioni CFD è un processo che deve prendere in considerazione più tipologie di esperimenti per fare un fit dei risultati in maniera da limitare gli errori provenienti da tutte le prove. Questo processo può diventare molto lungo e costoso se si prendono in considerazione prove sperimentali svolte nelle vasche apposite o attrezzando un prototipo reale in mare. Perciò le simulazioni sviluppate possono essere migliorate, in termini di discretizzazione spaziale e temporale soprattutto, tramite un con-

fronto con ulteriori dati provenienti da prove sperimentali trascurate durante il progetto svolto per la presente tesi, per questioni di tempo e costo. Questo processo rappresenta uno degli obiettivi futuri per quanto riguarda il percorso nell'ambito della costruzione degli strumenti per la progettazione delle imbarcazioni a vela tramite la fluidodinamica computazionale.

La necessità di rappresentare fedelmente la condizione di navigazione impone l'introduzione nel dominio delle onde. Infatti, nelle simulazioni svolte, il mare virtuale inserito nel modello VoF è un mare "piatto" privo di onde. Questa condizione è perlopiù una condizione teorica dato che, sia in lago che in mare è sempre presente il moto ondoso più o meno ampio.

L'introduzione dell'onda richiede un livello di discretizzazione della *free surface* maggiore e determinato dalle caratteristiche dell'onda, caratteristiche che comportano un aumento non trascurabile del costo computazionale. Per questo l'introduzione dell'onda rappresenta un ulteriore obiettivo futuro a completamento del set di simulazioni per la progettazione di una barca a vela tramite la fluidodinamica computazionale.

# Bibliography

- [1] <https://www.pinterest.it/pin/552042866800031785/>.
- [2] Corso di vela lezione 3: le andature. <https://blog.ormeggionline.com/velaandature/>.
- [3] Sillage (phénomène physique). [https://fr.wikipedia.org/wiki/Sillage\(ph](https://fr.wikipedia.org/wiki/Sillage(ph)
- [4] Pieter Rauwoens Björn Elsaesser Brecht Devolder, Pál Schmitt. A review of the implicit motion solver algorithm in openfoam to simulate a heaving buoy. 2015.
- [5] Santiago Marquez Damian. Description and utilization of interfoam multiphase solver. 2015.
- [6] Claudio Gallozzi Dario Dalla Vedova. I segreti della regata d'altomare. <https://besport.org/sportmedicina/velaregatealtomare.htm>.
- [7] Esi et al. Openfoam v2006. <https://develop.openfoam.com/Development/openfoam>.
- [8] Esi generated by doxygen. Openfoam: User guide v2006. <https://www.openfoam.com/documentation/guides>.
- [9] W. Malalasekera H. K. Versteeg. *An Introduction To Computational Fluid Dynamics*. 2018. Pearson Education, 2007.
- [10] Robert L. Street Joel H. Ferziger, Milovan Peric. *Computational Methods for Fluid Dynamics*. 2018. Springer, 2018.
- [11] Rolf E. Eliasson Lars Larsson. *Principles of Yacht Design*. 2000. Adlard Coles Nautical, 2000.
- [12] D.J. Mavriplis M.J. Brazell, J. Sitaraman. *An overset mesh approach for 3D mixed element high-order discretizations*. 2018. Pearson Education, 2007.

- [13] SimWiki. Cfd: Pimple algorithm.  
<https://www.simscale.com/forum/t/cfd-pimple-algorithm/81418>
- [14] Petra Tisovska. Description of the overset mesh approach in esi version of openfoam. 2019.
- [15] Classe velica. <https://it.wikipedia.org/wiki/classevelica>.