

# POLITECNICO DI TORINO

Ingegneria Chimica e dei Processi Sostenibili



Tesi di Laurea Magistrale

## Simulazione fluidodinamica del processo di riempimento di bevande gassate in bottiglia

Relatori

Prof. Antonio BUFFO

Prof. Daniele MARCHISIO

Dott. Giovanni TRONCI

Candidato

Roberto ARDUINO

Dicembre 2020



# Sommario

All'interno del lavoro di tesi viene studiato, mediante l'utilizzo della fluidodinamica computazionale, il processo di riempimento di bevande gassate in bottiglia. Il riempimento viene eseguito in pressione e il serbatoio del liquido è messo in comunicazione con la bottiglia mediante una valvola rotativa. Al fine di simulare l'operazione di riempimento, si è fatto uso del metodo Volume of Fluid (VOF) di tracciamento d'interfaccia per fluidi multifase. Esso è stato implementato in OpenFOAM mediante il `solver` InterFoam, molto sensibile rispetto alla griglia computazionale. Per questo motivo, all'interno della tesi sono stati effettuati molti tentativi di griglia, realizzate con l'ausilio di `SnappyHexMesh`, al fine di descrivere accuratamente il film liquido aderente alla parete della bottiglia. Inoltre, è stato realizzato, mediante `Paraview`, un metodo innovativo al fine di calcolare in ogni istante la frazione di gas intrappolata nella fase stagnante liquida in continuo accumulo nel fondo della bottiglia.

# Ringraziamenti

Ai miei genitori, Anna e Carlo.  
Grazie.



# Indice

|  |      |
|--|------|
| <b>Elenco delle tabelle</b>                                    | VIII |
| <b>Elenco delle figure</b>                                     | IX   |
| <b>1</b>   | 1    |
| 1.1 Introduzione . . . . .                                     | 1    |
| 1.2 Contesto applicativo . . . . .                             | 2    |
| 1.3 Obiettivi e Metodi . . . . .                               | 2    |
| <b>2 Aspetti Teorici</b>                                       | 5    |
| 2.1 Equazioni di governo . . . . .                             | 5    |
| 2.2 Tensione superficiale . . . . .                            | 7    |
| 2.3 Modello di turbolenza . . . . .                            | 12   |
| 2.4 Metodi per il tracciamento dell'interfaccia . . . . .      | 14   |
| 2.4.1 Metodi di superficie . . . . .                           | 15   |
| 2.4.2 Metodi di volume . . . . .                               | 18   |
| 2.4.3 Volume of Fluid Method . . . . .                         | 20   |
| 2.5 Modello matematico adottato nel solver InterFOAM . . . . . | 22   |
| 2.5.1 Equazioni del modello . . . . .                          | 25   |
| <b>3 Soluzione numerica delle equazioni del modello</b>        | 27   |
| 3.1 Metodo ai volumi finiti . . . . .                          | 27   |
| 3.2 Discretizzazione del dominio fluido . . . . .              | 28   |
| 3.3 Discretizzazione delle equazioni . . . . .                 | 29   |
| 3.3.1 Discretizzazione del termine di convezione . . . . .     | 30   |
| 3.3.2 Discretizzazione del termine di diffusione . . . . .     | 31   |
| 3.3.3 Discretizzazione del termine di generazione . . . . .    | 32   |
| 3.3.4 Discretizzazione del termine temporale . . . . .         | 33   |
| 3.3.5 Sistema algebrico risolutivo . . . . .                   | 34   |
| 3.4 Descrizione di OpenFOAM . . . . .                          | 34   |
| 3.4.1 Directory constant . . . . .                             | 35   |

|          |   |           |
|----------|---|-----------|
| 3.4.2    | Directory <code>system</code> . . . . .   | 36        |
| 3.4.3    | Directories <code>time</code> . . . . .   | 36        |
| 3.5      | Descrizione del solver <code>interFoam</code> . . . . .   | 37        |
| <b>4</b> | <b>Dettagli numerici e descrizione del caso studiato</b>  | <b>44</b> |
| 4.1      | Descrizione generale e piano di simulazioni . . . . .   | 44        |
| 4.1.1    | Regime fluido adottato all'interno delle simulazioni di valvola<br>e bottiglia . . . . .                    | 45        |
| 4.2      | Geometria e griglia finali del complesso valvola-bottiglia . . . . .  | 46        |
| 4.2.1    | Geometria e griglia finale: valvola . . . . .   | 46        |
| 4.2.2    | Geometria e griglia finale: bottiglia . . . . .   | 46        |
| 4.3      | Condizioni al contorno e iniziali riguardanti la simulazione della<br>valvola: cartella 0 . . . . .         | 47        |
| 4.3.1    | File <code>U</code> . . . . .   | 48        |
| 4.3.2    | File <code>p_rgh</code> . . . . .   | 49        |
| 4.3.3    | File <code>alpha.water.orig</code> . . . . .  | 50        |
| 4.4      | Condizioni al contorno e iniziali riguardanti la simulazione della<br>bottiglia: cartella 0 . . . . .       | 51        |
| 4.4.1    | File <code>U</code> . . . . .   | 51        |
| 4.4.2    | File <code>p_rgh</code> . . . . .   | 51        |
| 4.4.3    | File <code>alpha.water.orig</code> . . . . .  | 52        |
| 4.4.4    | File <code>nut</code> . . . . .   | 52        |
| 4.4.5    | File <code>epsilon</code> . . . . .   | 52        |
| 4.4.6    | File <code>k</code> . . . . .   | 52        |
| 4.5      | Dettagli riguardanti la simulazione della bottiglia: wall functions . .                                     | 52        |
| 4.6      | Schemi numerici adottati . . . . .  | 55        |
| 4.7      | Proprietà dei fluidi . . . . .  | 58        |
| 4.8      | Regime fluido all'interno delle simulazioni di valvola e bottiglia . . .                                    | 59        |
| 4.8.1    | Simulazione finale della valvola . . . . .  | 59        |
| 4.8.2    | Simulazione finale della bottiglia . . . . .  | 59        |
| 4.9      | Controlli di simulazione . . . . .  | 60        |
| <b>5</b> | <b>Creazione delle griglie e descrizione del film</b>   | <b>62</b> |
| 5.1      | Descrizione di una generica griglia computazionale in OpenFOAM .  | 62        |
| 5.2      | Creazione di una mesh in OpenFOAM . . . . .   | 63        |
| 5.2.1    | Geometria . . . . .   | 63        |
| 5.2.2    | <code>blockMesh</code> . . . . .  | 63        |
| 5.2.3    | <code>surfaceFeatureExtract</code> . . . . .  | 65        |
| 5.2.4    | <code>snappyHexMesh</code> . . . . .  | 65        |
| 5.3      | Descrizione del sottile film liquido alla parete: percorso di definizione<br>della griglia finale . . . . . | 71        |

|          |  |            |
|----------|--|------------|
| 5.3.1    | Griglie riguardanti la simulazione della valvola . . . . .   | 72         |
| 5.4      | Griglie riguardanti la simulazione della bottiglia . . . . .   | 73         |
| <b>6</b> | <b>Risultati</b>   | <b>89</b>  |
| 6.1      | Determinazione angolo di swirl e velocità tangenziale . . . . .  | 89         |
| 6.2      | Gas intrappolato all'interno del battente liquido: procedura e risultati   | 91         |
| 6.2.1    | Procedura per il calcolo del gas hold-up tramite Paraview .  | 91         |
| 6.2.2    | Gas hold-up: risultati . . . . .   | 92         |
| 6.3      | Analisi riguardante l'influenza di concentrazione, temperatura e<br>portata di riempimento sulla produzione di schiuma durante la<br>prima fase di degasamento . . . . . | 96         |
| 6.3.1    | Procedura di degasamento . . . . .   | 97         |
| 6.3.2    | Campagna sperimentale . . . . .  | 97         |
| 6.3.3    | Influenza dei fattori sul primo picco di schiuma: risultati . .  | 99         |
| 6.4      | Tentativo di ottenimento dell'altezza di schiuma nella prima fase di<br>degasamento tramite espansione del gas intrappolato . . . . .                                    | 102        |
| <b>7</b> | <b>Conclusioni</b>   | <b>104</b> |

# Elenco delle tabelle

|     |   |     |
|-----|---|-----|
| 2.1 | Coefficienti di chiusura Modello $k - \epsilon$ Standard. . . . .   | 14  |
| 4.1 | Tempo di riempimento attivo e riposo per le tre portate simulate. . .   | 45  |
| 4.2 | Proprietà fluide utilizzate nelle simulazioni finali di valvola e bottiglia   | 59  |
| 5.1 | Frazione di gas all'interno del battente liquido nel fondo della<br>bottiglia dopo la fase di riposo per le portate $Q_1$ e $Q_2$ . . . . . | 80  |
| 6.1 | Collezione di dati sperimentali sul primo picco di schiuma. . . . .   | 98  |
| 6.2 | Altezza di primo picco calcolata sulla base delle simulazioni effettuate.   | 102 |

# Elenco delle figure

|      |  |    |
|------|--|----|
| 1.1  | Fasi del processo di riempimento di bevande gassate . . . . .  | 3  |
| 2.1  | Porzione di interfaccia per la definizione di tensione interfacciale: l'elemento di interfaccia di area superficiale $A$ è stirato dalla forza distribuita $\gamma$ agente sul perimetro $l$ . . . . . | 7  |
| 2.2  | Tensione interfacciale in termini di lavoro reversibile: per estendere la superficie ogni elemento $dl$ del perimetro deve essere spostato di $dx$ da una forza $dF$ agente su di esso. . . . .        | 8  |
| 2.3  | Raggi di curvatura per una superficie spaziale $S$ ad un punto $P$ . . .   | 9  |
| 2.4  | Illustrazione dell'azione della tensione superficiale . . . . .  | 9  |
| 2.5  | CFS: area di transizione . . . . .   | 11 |
| 2.6  | CFS: rappresentazione della normale e della curvatura all'interfaccia  | 11 |
| 2.7  | Continuità delle velocità e discontinuità della quantità di moto nel flusso bifasico. . . . .  | 15 |
| 2.8  | Interface particles method. . . . .  | 16 |
| 2.9  | Height functions method. . . . .   | 17 |
| 2.10 | Level set method. . . . .  | 18 |
| 2.11 | Surface fitted method. . . . .   | 18 |
| 2.12 | Marker particles method. . . . .   | 19 |
| 2.13 | Volume fractions method. . . . .   | 20 |
| 3.1  | Esempio di volume di controllo in cui suddiviso il dominio fluido . .  | 28 |
| 3.2  | Interpolazione nel CDS . . . . .   | 31 |
| 3.3  | Minimum Correction Approach. . . . .   | 32 |
| 3.4  | Struttura della libreria OpenFOAM. . . . .   | 35 |
| 3.5  | Struttura di una generica directory base di simulazione in Open-FOAM. . . . .  | 36 |

|     |   |    |
|-----|---|----|
| 4.1 | Sinistra: viene rappresentata la geometria di simulazione della valvola al completo. In verde è evidenziata la sezione di <code>outlet</code> , dalla quale uscirà il gas in pressione contenuto all'interno della bottiglia a causa dell'entrata del liquido. In giallo vi è la sezione circolare di ingresso del film liquido: l' <code>inlet</code> . In blu sono indicate le pareti della valvola, denominate <code>Valve_walls</code> , mentre in rosso è rappresentata una parte della parete della bottiglia, denominata <code>Top_bottle</code> . Infine, in bianco è indicata la sezione inferiore della prima parte della bottiglia, chiamata all'interno della simulazione <code>bottomOutflow</code> . Destra: viene riportata una vista in sezione della valvola . . . . .   | 47 |
| 4.2 | Sinistra: sezione della zona centrale della mesh. Destra: particolare della sezione nella zona terminale della valvola. Si noti l'infittimento uniforme costituito da celle molto piccole . . . . .   | 48 |
| 4.3 | Sinistra: viene rappresentata la geometria di simulazione della bottiglia al completo. In giallo è presente la sezione di <code>outlet</code> , dalla quale uscirà il gas in pressione contenuto all'interno della bottiglia a causa dell'entrata del liquido. In verde vi è la sezione di circolare di ingresso del film liquido: l' <code>inlet</code> . Questa corona circolare è utilizzata al fine di avere già all'ingresso della bottiglia il film liquido attaccato alla parete, come accade nelle prove sperimentali. In blu viene indicata la parete <code>neck</code> , la quale è un artificio di simulazione la cui utilità viene descritta all'interno del capitolo 5. Infine, la parete della bottiglia è stata divisa in due sezione: in bianco <code>Top_bottle</code> e in rosso <code>Bottom_bottle</code> . Questa suddivisione della parete è finalizzata alla realizzazione degli strati di celle denominati <code>layers</code> che permettono di descrivere accuratamente il film liquido alla parete. Destra: sezione della griglia computazionale adottata. Si noti l'infittimento alle pareti e nella zona del tappo. Qui, infatti, è necessario descrivere accuratamente l'interfaccia gas-liquido. . . . | 49 |
| 4.4 | Particolare della zona del tappo all'interno della griglia computazionale della bottiglia. Si notino i 10 <code>layers</code> attaccati alla parete e la box d'infittimento. . . . .  | 50 |
| 4.5 | Relazione fra $U^+$ e $y^+$ all'interno delle tre regioni all'interno dell'inner region attaccata alla parete. . . . .  | 56 |
| 5.1 | Esempio di geometria in formato <code>.stl</code> . . . . .   | 64 |
| 5.2 | <code>blockMesh</code> applicato alla geometria della bottiglia . . . . .   | 65 |
| 5.3 | Diversi livelli di refinement a partire da una dimensione di cella base   | 67 |

|      |  |    |
|------|--|----|
| 5.4  | Castellated Mesh in una porzione della bottiglia. Si noti la seghet-<br>tatura presente lungo la superficie. Ciò è dato dalla non perfetta<br>aderenza della mesh alla geometria di riferimento. Inoltre, viene<br>effettuato un refinement delle celle del blocco di base . . . . .   | 68 |
| 5.5  | Snapped Mesh in una porzione della bottiglia. Si noti come ora la<br>mesh sia perfettamente aderente alla superficie della geometria. . .  | 69 |
| 5.6  | Aggiunta di sei strati di celle alla parete superiore della bottiglia. .   | 71 |
| 5.7  | Sinistra: taglio della griglia computazionale di prova Destra:portata<br>$Q_3$ , alpha.water, time step 1 sec. Si noti lo scollamento del film<br>liquido. . . . .   | 73 |
| 5.8  | Sinistra: portata $Q_3$ , alpha.water, time step 1sec. Si noti lo scolla-<br>mento del film liquido. Destra: taglio della griglia computazionale<br>finale . . . . .   | 74 |
| 5.9  | Sinistra: geometria implementata nel primo set di simulazioni. Si<br>noti come la bottiglia sia legata ad una piccola porzione di valvola.<br>Destra: griglia computazionale utilizzata all'interno del primo set di<br>simulazioni. Si noti la presenza di una box infittita nella zona del<br>tappo della bottiglia e l'infittimento generale alle pareti . . . . .  | 75 |
| 5.10 | Vettori velocità enormi fra il confine delle sezioni di <b>inlet</b> ed <b>outlet</b>  | 75 |
| 5.11 | Inserimento di una parete buffer. In giallo si indica l' <b>outlet</b> , ovvero<br>la sezione di fuoriuscita del gas, in verde l' <b>inlet</b> , ovvero la sezione<br>di ingresso della portata liquida e in rosso la parete buffer che separa<br>le due sezioni. . . . .  | 76 |
| 5.12 | Sinistra: è possibile osservare la geometria adottata, nella quale<br>vi è un allontanamento delle sezioni di <b>inlet</b> ed <b>outlet</b> . Destra:<br>vengono mostrati i vettori velocità derivanti da una delle simulazioni.<br>Si noti come ora i vettori presenti al congiungimento delle sezioni di<br><b>inlet</b> ed <b>outlet</b> siano dello stesso ordine di grandezza per quanto<br>riguarda il modulo dei vettori interni al campo di simulazione. . . . | 76 |
| 5.13 | Nell'immagine viene mostrata la griglia computazionale nella zone<br>di <b>inlet</b> e i vettori velocità. Inoltre è possibile osservare una certa<br>diffusione del liquido all'interno della fase gassosa. . . . .   | 77 |
| 5.14 | Sinistra: anello di gas intrappolato ad ingresso bottiglia nei primi<br>istanti di simulazione. Destra: scomparsa dell'anello di gas nei time<br>step successivi e perfetta aderenza alla parete del film liquido ad<br>ingresso bottiglia . . . . .   | 78 |
| 5.15 | Sinistra: si noti il precoce scollamento del film liquido alla parete<br>della bottiglia. Destra: vortici di gas nella porzione terminale della<br>valvola. . . . .  | 79 |

|      |   |    |
|------|---|----|
| 5.16 | Sinistra: geometria contenente una porzione maggiore di valvola. Destra: taglio di parte della griglia computazionale. Si noti la presenza di strati di celle alle pareti della bottiglia e l'infittimento generale nella zona della valvola. . . . .   | 80 |
| 5.17 | Visibile distacco del film liquido anche con la nuova geometria e griglia adottate. . . . .   | 81 |
| 5.18 | Sinistra: evidente scollamento del film liquido a parete, comunque minore rispetto a quello ottenuto nel set di simulazioni precedente. Destra: nel taglio vengono evidenziate varie zone di diffusione del film liquido all'interfaccia. . . . .   | 82 |
| 5.19 | Taglio della griglia computazionale nella zona superiore. Si noti come la presenza di una box raffinata e dei layers deformi la griglia e la vada a comprimere. Inoltre, con il trattino viola si indica il diametro della sezione di <i>inlet</i> : si evidenzia dunque la volontà di inglobare il film liquido all' interno della porzione raffinata. . . . . | 83 |
| 5.20 | Sinistra: taglio della griglia computazionale nella zona superiore. Si noti come essa è molto meno deformata rispetto al tentativo precedente. Destra: lo scollamento del film liquido è ancora presente. Si ha, però, un progressivo miglioramento. . . . .  | 83 |
| 5.21 | Sinistra: taglio della griglia computazionale nella zona superiore. Si noti come 6 layers e l'assenza di infittimento non permettano una corretta descrizione del fenomeno diffusivo d'interfaccia. Destra: scollamento del film liquido dalla parete nonostante le modifiche. . .  | 84 |
| 5.22 | Sinistra: taglio della griglia computazionale nella parte superiore. Destra: notevole miglioramento riguardante il problema dello scollamento del film liquido. . . . .   | 85 |
| 5.23 | Punti di distacco dell'interfaccia. Si vede come il distacco avvenga dove il film non è completamente contenuto nello strato di layers. . . . .   | 85 |
| 5.24 | Sinistra: taglio della griglia computazionale nella parte superiore. Si noti come le zone diffusive non vengano descritte bene. Destra: ulteriore miglioramento riguardante il problema dello scollamento del film liquido. . . . .   | 86 |
| 5.25 | Sinistra: taglio della griglia computazionale nella parte superiore. Si noti come vi sia una certa deformazione della griglia. Destra: importante peggioramento riguardante il problema dello scollamento del film liquido. Diventa sempre più palese l'importanza di avere una griglia uniforme . . . . .  | 87 |

|      |  |     |
|------|--|-----|
| 6.1  | Portata $Q_1$ , <code>alpha.water</code> , time step 1 secondo: due diversi posizionamenti del comando <code>Slice</code> affinché si ottenga la sezione di taglio voluta. . . . .   | 90  |
| 6.2  | Portata $Q_1$ , <code>alpha.water</code> , time step 0,5 secondi. Sinistra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo pre- <code>Threshold</code> . Destra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo post- <code>Threshold</code> . . . . . | 92  |
| 6.3  | Portata $Q_1$ , <code>alpha.water</code> , time step 1 secondo. Sinistra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo pre- <code>Threshold</code> . Destra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo post- <code>Threshold</code> . . . . .   | 93  |
| 6.4  | Portata $Q_1$ , <code>alpha.water</code> , time step 6 secondi. Sinistra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo pre- <code>Threshold</code> . Destra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo post- <code>Threshold</code> . . . . .   | 93  |
| 6.5  | Portata $Q_1$ : andamento della frazione di gas all'interno del battente liquido. . . . .  | 94  |
| 6.6  | Portata $Q_2$ : andamento della frazione di gas all'interno del battente liquido. . . . .  | 94  |
| 6.7  | Portata $Q_3$ : andamento della frazione di gas all'interno del battente liquido. . . . .  | 95  |
| 6.8  | Andamento della frazione gassosa intrappolata nel battente liquido per le tre portate all'interno della fase di riposo. . . . .  | 95  |
| 6.9  | Frazione di gas intrappolata nel battente liquido prevista interpolando i valori ottenuti dalle simulazioni delle portate $Q_1$ , $Q_2$ e $Q_3$ . . . . .  | 96  |
| 6.10 | La curva nera è una rappresentazione approssimata del profilo di schiuma, il quale può essere scomposto idealmente nella curva verde, rappresentante il contributo del riempimento, e nella curva rossa, rappresentante il contributo della seconda fase di degasamento. . . . .                         | 98  |
| 6.11 | Altezza di schiuma ottenuta nella prima di fase di degasamento variando la concentrazione e mantenendo costanti la portata e la temperatura. . . . .   | 99  |
| 6.12 | Altezza di schiuma ottenuta nella prima di fase di degasamento variando la portata e mantenendo costanti la concentrazione e la temperatura. . . . .   | 100 |
| 6.13 | Altezza di schiuma ottenuta nella prima di fase di degasamento variando la temperatura e mantenendo costanti la concentrazione e la portata. . . . .   | 100 |
| 6.14 | MODDE: <code>Coefficients Plot</code> . . . . .  | 101 |
| 6.15 | MODDE: <code>Vip plot</code> . . . . .   | 101 |

|      |   |     |
|------|---|-----|
| 6.16 | Sinistra: time step corrispondente alla fine del riposo di 0.3 secondi per una delle tre portate. Destra: l'applicazione del comando <code>clip</code> permette l'ottenimento di una porzione di volume di spessore 1 cm situata sotto l'interfaccia. . . . . | 103 |
|------|---|-----|



# Capitolo 1

## 1.1 Introduzione

Al giorno d'oggi, l'ampia varietà di prodotti incontrati nel mercato delle bevande riflette un background di intensa creatività in termini di caratterizzazione dei tipi di bevanda. L'area delle bibite gassate non fa eccezione. L'arte di produrre "carbonated soft drinks" si è sviluppata in più di duecento anni, nei quali l'idea originale venne dall'osservazione dell'acqua frizzante naturale. Essa è prodotta naturalmente grazie alla presenza di anidride carbonica in una falda acquifera, all'interno della quale l'alta pressione ne consente l'assorbimento in acqua. Nel 1767, Joseph Priestley produce acqua frizzante artificialmente. Egli scoprì che sospendendo una caraffa d'acqua all'interno di un serbatoio di birra vi era dissoluzione di anidride carbonica all'interno dell'acqua. Nell'anno 1783, Jacob Schwepes inventò un sistema di produzione di acqua gassata ad alta efficienza, che consentì l'inizio di una commercializzazione su larga scala del prodotto. Negli anni seguenti, l'acqua gassata divenne popolare in tutto il mondo, soprattutto grazie alla sensazione di freschezza che conferiva, e, al fine di espandere sempre di più la scala commerciale, si iniziò ad aggiungere all'interno di essa sciroppi e additivi. Fu l'inizio del mercato delle bibite gassate, nel quale le caratteristiche organolettiche del prodotto e l'efficienza di produzione sono fortemente influenzate dallo sviluppo tecnologico dei materiali, dalla tecnica di carbonatazione e dal processo di riempimento dei contenitori adibiti al trasporto. Da questa premessa emerge come la comprensione e lo studio dei sistemi gas-liquido dal punto di vista termodinamico e fluidodinamico sia fondamentale nell'industria alimentare per l'ottenimento di prodotto ad alto valore aggiunto e competitivo sul mercato.

---

## 1.2 Contesto applicativo

In questo lavoro di tesi viene studiato il processo di riempimento di bevande gassate. Il lavoro è stato svolto in collaborazione con SIDEL s.p.a, azienda leader nel settore della fornitura di soluzioni in PET per l'imbottigliamento di liquidi, e Optimad engineering Srl, società che opera nel campo della produzione e commercializzazione di software per l'analisi, la progettazione e l'ottimizzazione fluidodinamica.

All'interno dell'industria alimentare, il termine carbonatazione intende il processo tramite il quale avviene la dissoluzione di  $CO_2$  in acqua. Esso dipende fortemente da pressione e temperatura.

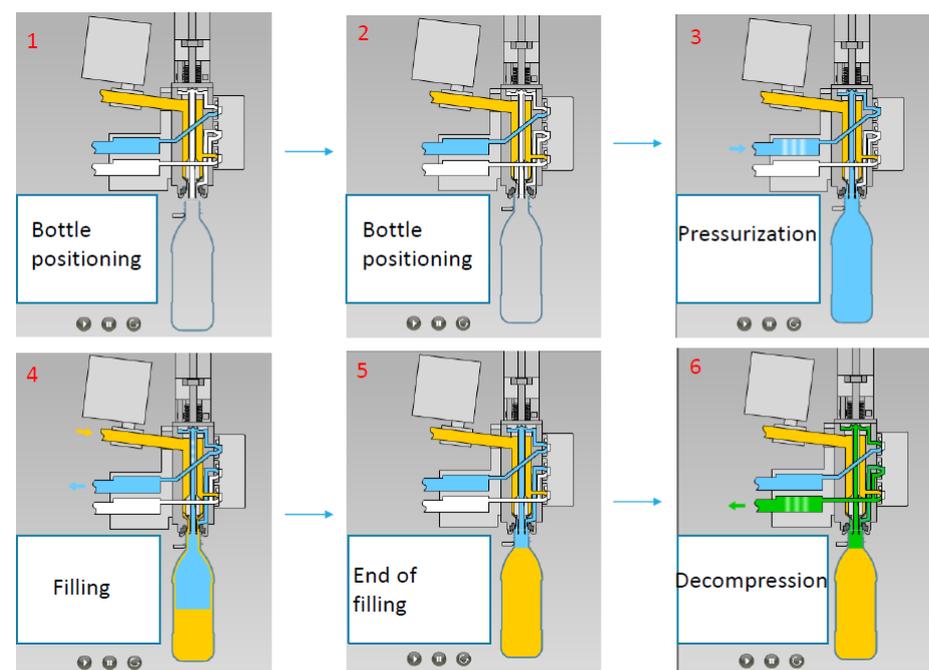
A seguito della carbonatazione la bibita viene imbottigliata. Il processo di imbottigliamento è costituito da diverse fasi in sequenza:

- posizionamento della bottiglia
- pressurizzazione della bottiglia a pressione  $P_{tank}$  con  $CO_2$
- inserimento del liquido e contemporanea fuoriuscita del gas a pressione costante
- depressurizzazione della bottiglia ad 1 bar e conseguente chiusura della stessa tramite tappo.

Le fasi costituenti il processo di riempimento sono illustrate all'interno della Figura 1.1. Durante la fase di inserimento del liquido parte del gas viene catturato all'interno della fase liquida. Successivamente, durante la fase di depressurizzazione vi è la possibile produzione di schiuma, la quale deve essere ridotta il più possibile al fine di non interferire con il processo di riempimento. Infatti, essa durante la fase di degasamento potrebbe fuoriuscire dalla bottiglia e causare, dunque, perdita di liquido lungo la linea produttiva e sporcando la parete esterna della bottiglia. La produzione di schiuma è influenzata da molti fattori, fra cui la concentrazione di gas all'interno del liquido in bottiglia. Nel processo di riempimento della bottiglia, la valvola rotativa di collegamento fra il serbatoio e la bottiglia riveste un ruolo fondamentale. Essa, infatti, imprime al liquido un moto rotatorio (swirl) lungo l'asse verticale: in questo modo si crea un film di liquido aderente alla parete della bottiglia, consentendo un più efficace e veloce riempimento. Lo swirl, in aggiunta, ha anche al funzione di ridurre l'intrappolamento di bolle di gas nel liquido, affinché si riduca la produzione di schiuma.

## 1.3 Obiettivi e Metodi

All'interno del lavoro di tesi si è fatto uso della fluidodinamica computazionale al fine di simulare il riempimento di una bottiglia di bevanda gassata. Il processo



**Figura 1.1:** Fasi del processo di riempimento di bevande gassate

di riempimento viene effettuato inserendo il liquido all'interno di una bottiglia di PET da 1,5 litri, di comune impiego all'interno del packaging alimentare. L'obiettivo principale è stato lo sviluppo e la creazione di un metodo semplice in post-processing per il monitoraggio del gas intrappolato all'interno della fase liquida nel tempo, definito in inglese come "hold-up". Nella seconda parte della tesi, invece, l'attenzione è stata rivolta ad individuare una relazione, tramite metodologia statistica Design of Experiment, fra l'hold-up di gas all'interno del liquido a fine riempimento e il comportamento del sistema durante la prima parte di degasamento. Per realizzare ciò si è fatto uso, tramite il sistema operativo Unix-like Ubuntu, del toolbox C++ OpenFOAM, il quale ha permesso di risolvere, tramite procedure di calcolo numerico, le equazioni caratterizzanti un sistema di due fluidi immiscibili e isotermi. Il processo di simulazione ha richiesto l'utilizzo di risorse di calcolo ad alte prestazioni, le quali sono state fornite da HPC@POLITO ACADEMIC COMPUTING CENTER. La sfida principale all'interno di questo lavoro è stata l'ottenimento di una simulazione che rispecchiasse la fisica reale osservata sperimentalmente nel processo d'imbottigliamento. Queste difficoltà incontrate, come verrà ampiamente discusso nella trattazione, sono emerse a causa della forte dipendenza del metodo di tracciamento dell'interfaccia utilizzato nei confronti della discretizzazione spaziale del dominio fluido e nei confronti del regime di flusso derivante dalle condizioni operative investigate. All'interno di questo lavoro, per ottenere

---

la discretizzazione spaziale del dominio fluido si è fatto uso dell'utilty di meshing open-source snappyHexMesh e di STAR-CCM+. La procedura di monitoraggio del gas hold-up e la visualizzazione dei risultati di simulazione sono stati possibili mediante l'impiego dell'applicazione open-source Paraview. Infine, animazioni ad alta risoluzione sono state ottenute mediante il software libero di montaggio video Blender.

# Capitolo 2

## Aspetti Teorici

### 2.1 Equazioni di governo

Nell'ambito della fluidodinamica computazionale (CFD) vengono risolte equazioni di bilancio di materia, quantità di moto ed energia. La soluzione analitica di queste equazioni differenziali non lineari alle derivate parziali è disponibile solo in pochi casi limite e, per questa ragione, vengono usate tecniche numeriche al fine di ottenerne la soluzione.

In generale, l'equazione di bilancio in notazione indiciale per uno scalare, vettore o tensore  $\phi$  si presenta nella forma

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \vec{U} \phi) = \nabla \cdot (\rho \Gamma_\phi \nabla \phi) + S_\phi(\phi) \quad (2.1)$$

dove i termini hanno il seguente significato:

$$\{\text{Accumulo}\} + \{\text{Trasporto convettivo}\} = \{\text{Trasporto diffusivo}\} + \{\text{Generazione}\}$$

Si consideri un generico fluido per cui sia applicabile la meccanica del continuo, ovvero per il quale un'accurata descrizione possa essere realizzata senza considerare il moto molecolare. L'ipotesi di continuità prevede che il rapporto fra il cammino libero medio molecolare  $\lambda$  e la lunghezza caratteristica  $L$  sia molto minore di uno, ovvero  $\text{Kn} = \frac{\lambda}{L} \ll 1$ , dove  $\text{Kn}$  viene definito numero di Knudsen. Si consideri, inoltre, il comportamento newtoniano del fluido. Per fluido newtoniano si intende un fluido caratterizzato da un coefficiente di viscosità indipendente dalla velocità di deformazione e costante. Per un fluido di questo tipo il tensore degli sforzi è scritto come:

$$\Sigma = \mu(\nabla \vec{U} + (\nabla \vec{U})^T) - \left(\frac{2}{3}\mu - \kappa\right)\mathbf{I}(\nabla \cdot \vec{U}) - \mathbf{I}P. \quad (2.2)$$

Per un fluido qualunque è bene ricordare che il tensore degli sforzi si può esprimere come

$$\boldsymbol{\Sigma} = -P\mathbf{I} + \mathbf{D} \quad (2.3)$$

dove  $\mathbf{D}$  è il tensore deviatorico a traccia nulla, mentre per quanto riguarda la pressione  $P = -\frac{1}{3}\text{Tr}(\boldsymbol{\Sigma})$ .

La viscosità dilatazionale  $\kappa$  è importante solo nei casi di onda d'urto o di onde sonore. Dalla teoria cinetica dei gas è noto che  $\kappa$  è nulla per gas monoatomici a bassa pressione.

Ora, a seconda dell'equazione di riferimento quanto riportato in (2.1) diventa: bilancio di materia, quantità di moto ed energia. Detto ciò, per un fluido incomprimibile è inoltre possibile semplificare l'equazione di continuità come

$$\nabla \cdot \vec{U} = 0. \quad (2.4)$$

Dunque, per un fluido newtoniano e incomprimibile è possibile scrivere il tensore deviatorico di flusso di quantità di moto come

$$\mathbf{T} = -\mathbf{D} = -\mu(\nabla\vec{U} + (\nabla\vec{U})^T). \quad (2.5)$$

Considerando le assunzioni fatte in precedenza, ed aggiungendo ad esse la condizione di temperatura costante del fluido è possibile, dunque, trascurare il bilancio termico al fine di descrivere il comportamento fluido. Risultano necessarie solamente due equazioni:

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\vec{U}) = 0. \quad (2.6)$$

$$\frac{\partial\vec{U}}{\partial t} + \vec{U} \cdot (\nabla\vec{U}) = -\frac{1}{\rho}\nabla P - \frac{1}{\rho}(\nabla \cdot \mathbf{T}) + \vec{f}. \quad (2.7)$$

Se il fluido è newtoniano e incomprimibile, allora il bilancio di quantità di moto si traduce nell'equazione di Navier-Stokes, riportata qui sotto:

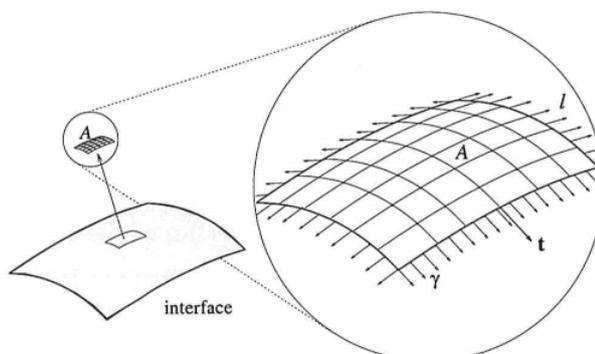
$$\frac{\partial\vec{U}}{\partial t} + \vec{U} \cdot (\nabla\vec{U}) = -\frac{1}{\rho}\nabla P + \nu\nabla \cdot (\nabla\vec{U} + (\nabla\vec{U})^T) + \vec{f}. \quad (2.8)$$

Quest'ultima può essere scritta in forma differente grazie al fatto che

$$\nu\nabla \cdot (\nabla\vec{U} + (\nabla\vec{U})^T) = \nu\Delta\vec{U} \quad (2.9)$$

vale nell'ipotesi di fluidi con  $\rho$  e  $\nu$  costanti. Nelle equazioni (2.7) (2.8) compare il termine  $\vec{f}$ . Esso rappresenta le forze esterne di volume applicate al fluido e solitamente esse si traducono nella sola forza di gravità e, dunque,  $\vec{f} = \vec{g}$ . In aggiunta alla gravità, ci possono essere campi esterni aggiuntivi che possono modificare l'accelerazione del fluido, ad esempio campi elettrici e magnetici. Nel caso di presenza di più fasi all'interno del dominio fluido è necessario includere la forza derivante dalla tensione superficiale  $\vec{f}_\gamma$  come contributo aggiuntivo al bilancio di quantità di moto.

## 2.2 Tensione superficiale



**Figura 2.1:** Porzione di interfaccia per la definizione di tensione interfacciale: l'elemento di interfaccia di area superficiale  $A$  è stirato dalla forza distribuita  $\gamma$  agente sul perimetro  $l$ .

Nella letteratura scientifica, il termine "tensione superficiale" è riservato alla tensione osservata in un'interfaccia liquido-vapore, mentre il termine "tensione interfacciale" è usato per riferirsi ad interfacce di ogni natura, includendo, ad esempio, l'interfaccia liquido-liquido.

In natura, un'interfaccia presenta la tendenza a minimizzare la propria area superficiale al fine di aumentare il numero di molecole localizzate nel *bulk* fluido, dove ogni molecola è circondata da un grande numero di molecole confinanti. Questa tendenza risalta agli occhi dell'osservatore soprattutto quando si è in presenza di interfacce liquide. Infatti, questa tendenza a contrarsi induce il liquido a comportarsi come se avesse una vera e propria "pelle".

La tendenza a contrarsi di un'interfaccia può essere quantificata modellando il sistema attraverso la tensione interfacciale  $\gamma$ .

Si consideri, ora, un sistema in equilibrio meccanico, dove vi è bilancio di tutte le forze, e si esamini una porzione di interfaccia di area superficiale  $A$  e di perimetro  $l$  (come mostrato in Figura 2.1).

Questa porzione di interfaccia è stirata dal resto dell'interfaccia attraverso una forza distribuita che agisce lungo tutto il perimetro e bilancia la tendenza a contrarsi dell'interfaccia. Dunque, è possibile definire la forza agente sulla porzione di perimetro  $dl$  come

$$dF_{eq} = \gamma dl \quad (2.10)$$

dove  $\gamma$  è la tensione interfacciale. La forza è pura tensione e di conseguenza in ogni punto del perimetro è orientata lungo il vettore  $\vec{t}$ , perpendicolare al perimetro e

tangente all'interfaccia. Questo è comunemente descritto come modello di membrana di Young, secondo il quale  $\gamma$  dipende dalla composizione, dalla pressione e dalla temperatura, ma non dalla forma della superficie.

Una diversa definizione di  $\gamma$  si basa sul lavoro necessario ad incrementare l'area superficiale di una interfaccia (Figura 2.2). Al fine di estendere la superficie di area  $A$  ad un'area  $A + dA$  è necessario muovere la superficie di contorno di  $dx$ . Dunque, visto che all'equilibrio è presente una forza  $dF_{eq}$  agente sul perimetro della membrana, ora sarà necessario applicare una forza  $dF > dF_{eq}$

$$dW = \oint_l(dx dF) > \oint_l(dx dF_{eq}) = \gamma \oint_l(dx dl) = \gamma dA. \quad (2.11)$$

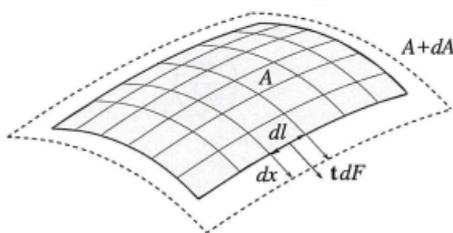
Inoltre, è possibile affermare che  $\gamma$  può essere definita in termini di lavoro necessario per un'espansione termodinamicamente reversibile dell'area superficiale di membrana. Dunque,

$$dW_{rev} = \gamma dA. \quad (2.12)$$

Ora, considerando un'interfaccia liquida curva, è possibile dire che vi è un salto di pressione attraverso l'interfaccia e la pressione risulta essere maggiore nella parte concava rispetto alla parte convessa. Questo concetto viene espresso dall'equazione di Young-Laplace, la quale lega il salto di pressione  $\Delta P$  tra le due fasi in un certo punto dell'interfaccia alla tensione interfacciale e alla curvatura locale

$$\Delta P = \gamma \left( \frac{1}{R_1} + \frac{1}{R_2} \right) = \gamma \kappa \quad (2.13)$$

dove  $R_1$  e  $R_2$  sono i due principali raggi di curvatura della superficie. Si definisce raggio di curvatura ad un generico punto  $P$  di una superficie curva il raggio del cerchio che nel punto  $P$  ha la stessa derivata prima e seconda della curva. Per definire la curvatura di una superficie  $S$  nello spazio, è necessario individuare il vettore normale  $\vec{n}$  nel punto di interesse  $P$  e far passare per esso due piani ortogonali,



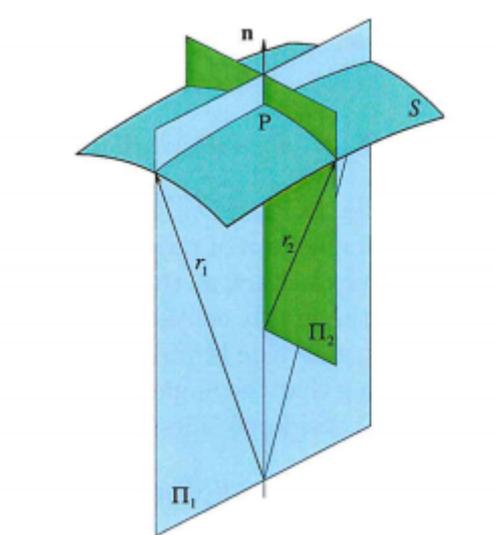
**Figura 2.2:** Tensione interfacciale in termini di lavoro reversibile: per estendere la superficie ogni elemento  $dl$  del perimetro deve essere spostato di  $dx$  da una forza  $dF$  agente su di esso.

come mostrato in Figura 2.3. Questi piani tagliano la superficie definendo due linee curve. A questo punto, è possibile definire la curvatura media alla superficie come:

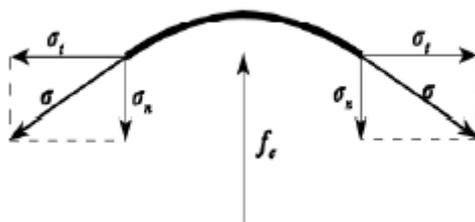
$$\kappa = \left( \frac{1}{r_1} + \frac{1}{r_2} \right). \quad (2.14)$$

Ora, esiste una orientazione dei piani per cui simultaneamente si ha che  $r_1$  è il massimo raggio di curvatura possibile e  $r_2$  è il minimo raggio di curvatura possibile. In questo caso l'orientazione dei piani è definita principale e i due raggi  $R_1$  e  $R_2$  vengono definiti raggi di curvatura principali.

Ora, nel caso di interfaccia curva, come si può vedere all'interno della Figura 2.4,



**Figura 2.3:** Raggi di curvatura per una superficie spaziale  $S$  ad un punto  $P$ .



**Figura 2.4:** Illustrazione dell'azione della tensione superficiale

la tensione superficiale non ha solo una componente tangenziale, ma anche una

componente normale. Se i due fluidi sono in equilibrio, siccome la componente tangenziale  $\sigma_t$  è bilanciata da se stessa, deve esistere una forza normale  $\vec{f}_\gamma$ , indicata in Figura 2.4 come  $\vec{f}_\sigma$ , causata dal salto di pressione, tale per cui si abbia bilancio con la componente verticale della tensione superficiale  $\sigma_n$ , come rappresentato in Figura 2.4. D'altro canto, se non vi fosse bilancio fra la forza netta derivante dal salto di pressione e la componente verticale della tensione superficiale allora la superficie libera inizierebbe a muoversi. Da ciò emerge come la forza di tensione superficiale sia strettamente collegata al salto di pressione attraverso l'interfaccia  $\vec{f}_\gamma = \nabla P$ . Tenendo presente ciò, l'equazione di Young-Laplace e che la forza di tensione superficiale agisce solo sulla superficie libera è possibile esprimere il tutto in termini matematici

$$\vec{f}_\gamma = \int_{s(t)} \gamma \kappa_{f_s} n_{f_s} \delta(\vec{x} - \vec{x}_{f_s}) dS \quad (2.15)$$

dove  $\kappa_{f_s}$ ,  $n_{f_s}$  e  $\vec{x}_{f_s}$  indicano rispettivamente la curvatura media, il vettore normale e il vettore posizione della superficie libera. Inoltre

$$\delta(\vec{x} - \vec{x}_{f_s}) = \begin{cases} 1 & \text{se } \vec{x} = \vec{x}_{f_s} \text{ il punto è sulla superficie libera,} \\ 0 & \text{se } \vec{x} \neq \vec{x}_{f_s} \text{ il punto non è sulla superficie libera.} \end{cases} \quad (2.16)$$

Ora, è possibile includere la forza derivante dalla tensione superficiale al bilancio di quantità di moto, come segue

$$\frac{\partial(\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \vec{U}) = -\nabla P - \nabla \cdot \tau + \rho \vec{g} + \vec{f}_\gamma. \quad (2.17)$$

Ora, il modello utilizzato per descrivere la forza di tensione superficiale presenta due problemi. Il primo luogo, vi è una discontinuità attraverso il dominio fluido dovuto al termine descritto in (2.16) e ciò potrebbe creare alcune difficoltà nella valutazione dell'integrale. In secondo luogo,  $\vec{f}_\gamma$  dipende fortemente dalla curvatura e dunque dalla forma dell'interfaccia e dalla posizione che è sconosciuta perché non è tracciata esplicitamente nel Metodo dei Volumi di Fluido, come verrà spiegato in seguito. Questi due problemi vengono risolti introducendo il modello di forza superficiale continua o "continuous surface force model" (CFS) della forza di tensione superficiale. In questo modello  $\vec{f}_\gamma$  è considerata come una forza di volume continua agente solo nella regione di transizione, ovvero nella regione dove la funzione indicatrice  $\alpha$  varia in modo regolare fra 0 e 1.

Nel CFS si ha che

$$\vec{f}_\sigma = \int_{s(t)} \sigma \kappa_{f_s} n_{f_s} \delta(\vec{x} - \vec{x}_{f_s}) dS \approx \sigma \kappa \vec{n} \quad (2.18)$$

$$\vec{n} = \nabla \alpha \quad (2.19)$$

$$\kappa = -\nabla \cdot \left( \frac{\vec{n}}{|\vec{n}|} \right). \quad (2.20)$$

Dunque, si definisce il vettore normale ai layers nella regione di transizione come gradiente della funzione indicatrice. La curvatura, invece, è definita come divergenza del versore normale.

Con questo modello è così possibile calcolare il contributo dovuto dalla tensione interfacciale nel bilancio di quantità di moto. Come si può vedere nell'equazione (2.18), questo contributo è diverso da 0 solamente in prossimità dell'interfaccia, in quanto il termine  $\vec{n}$  è non nullo solamente in prossimità di essa.

In conclusione, all'interno della Figura 2.5 è possibile vedere come la frazione di fase  $\alpha$  vari progressivamente fra 0 e 1 all'interno della regione di transizione, mentre all'interno della Figura 2.6 è riportata una rappresentazione schematica del risultato del calcolo del vettore normale e della curvatura.

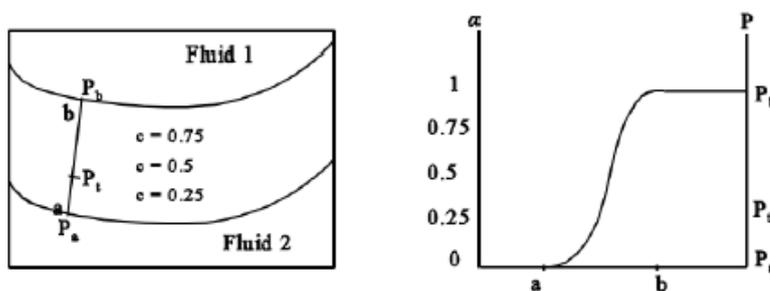


Figura 2.5: CFS: area di transizione

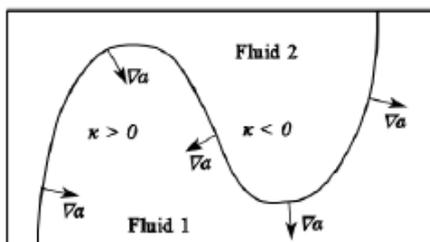


Figura 2.6: CFS: rappresentazione della normale e della curvatura all'interfaccia

## 2.3 Modello di turbolenza

All'interno delle simulazioni di applicazioni industriali, vengono impiegati alcuni modelli di turbolenza. Alcuni di essi rientrano nella categoria RANS ovvero "Reynolds Averaged Navier-Stokes" e si basano sulla decomposizione di Reynolds. Reynolds, più di 100 anni fa, espresse che variabili istantanee possono essere scritte come somma di un termine medio e di un termine fluttuante. Dunque, è possibile scrivere che

$$\vec{U} = \langle \vec{U} \rangle + \vec{u} \quad (2.21)$$

$$P = \langle P \rangle + p. \quad (2.22)$$

Con la decomposizione di Reynolds, il flusso è descritto statisticamente tramite una quantità mediata e una quantità fluttuante. Infatti, applicando la media ad un ragionevole periodo di tempo le fluttuazioni turbolente possono essere separate dalle quantità non turbolente. Si definisce media di Reynolds l'integrale

$$\langle \phi \rangle = \frac{1}{\tau} \int_t^{t+\tau} \phi(x, \tilde{t}) d\tilde{t}. \quad (2.23)$$

Da ciò deriva il fatto che la media di una quantità fluttuante è nulla, ovvero

$$\langle \vec{u} \rangle = 0. \quad (2.24)$$

Ora, sostituendo la forma decomposta delle variabili  $\vec{U}$  e  $P$  all'interno delle equazioni (2.4) e (2.8), privata del termine  $\vec{f}$ , e applicando ad esse la media di Reynolds si ottiene l'equazione di continuità e il bilancio di quantità di moto in campo turbolento

$$\nabla \cdot \langle \vec{U} \rangle = 0 \quad (2.25)$$

$$\frac{\partial \langle \vec{U} \rangle}{\partial t} + \langle \vec{U} \rangle \cdot \nabla \langle \vec{U} \rangle + \langle \nabla \cdot (\vec{u}\vec{u}) \rangle = -\frac{1}{\rho} \nabla \langle P \rangle + \nu \Delta \langle \vec{U} \rangle. \quad (2.26)$$

Con un riarrangiamento si ottiene la forma generale dell'equazione RANS

$$\frac{\partial \langle \vec{U} \rangle}{\partial t} + \langle \vec{U} \rangle \cdot \nabla \langle \vec{U} \rangle = -\frac{1}{\rho} \nabla \langle P \rangle + \nu \Delta \langle \vec{U} \rangle - \nabla \cdot \langle \vec{u}\vec{u} \rangle \quad (2.27)$$

che può essere scritta nella forma

$$\frac{\partial \langle \vec{U} \rangle}{\partial t} + \langle \vec{U} \rangle \cdot \nabla \langle \vec{U} \rangle = -\frac{1}{\rho} \nabla \cdot \{ \langle P \rangle \mathbf{I} + \mu (\nabla \langle \vec{U} \rangle + (\nabla \langle \vec{U} \rangle)^T) - \rho \langle \vec{u}\vec{u} \rangle \} \quad (2.28)$$

dove  $-\rho \langle \vec{u}\vec{u} \rangle$  è definito tensore degli sforzi di Reynolds (tensore di secondo ordine) ed è l'unico termine che deve essere modellato al fine di ottenere la "chiusura"

dell'equazione (2.27). Esso rappresenta il trasporto di quantità di moto dovuto alla turbolenza.

Una semplice approssimazione per avere la chiusura della RANS è l'approssimazione di Boussinesq, secondo la quale i componenti del tensore degli sforzi di Reynolds sono proporzionali ai gradienti di velocità media. La relazione di Boussinesq propone che il trasporto di quantità di moto da parte della turbolenza sia un processo diffusivo e che il tensore di Reynolds si possa modellare tramite una viscosità turbolenta (eddy viscosity), che è analoga alla viscosità molecolare. Dunque, si può scrivere che

$$\frac{\mathbf{T}}{\rho} = -\langle \vec{u}\vec{u} \rangle = \nu_t(\nabla\langle \vec{U} \rangle + (\nabla\langle \vec{U} \rangle)^T) - \frac{2}{3}k\mathbf{I}. \quad (2.29)$$

In aggiunta, è possibile introdurre la relazione

$$\mathbf{S} = \frac{1}{2}(\nabla\langle \vec{U} \rangle + (\nabla\langle \vec{U} \rangle)^T) \quad (2.30)$$

dove  $\mathbf{S}$  è il tensore velocità di deformazione.

All'interno dell'equazione (2.29)  $k$  è l'energia cinetica turbolenta per unità di massa.  $k$  è definita come

$$k = \frac{1}{2}\langle \vec{u}\vec{u} \rangle. \quad (2.31)$$

Al contrario della viscosità molecolare, la viscosità turbolenta non è una proprietà del fluido, ma dipende fortemente dalla magnitudo della turbolenza.

Il secondo termine a destra nella (2.29) rappresenta una tensione normale. Dunque, è possibile scrivere la (2.27) come

$$\frac{\partial\langle \vec{U} \rangle}{\partial t} + \langle \vec{U} \rangle \cdot \nabla\langle \vec{U} \rangle = -\frac{1}{\rho}\nabla(\langle P \rangle + \frac{2}{3}\rho k) + \nabla \cdot [(\nu + \nu_t)\nabla\langle \vec{U} \rangle]. \quad (2.32)$$

Il modello di Boussinesq è limitato: può predire solo flussi isotropici in equilibrio locale. Al fine di risolvere la (2.32) è necessario determinare la viscosità turbolenta  $\nu_t$ . Spesso, quando si tratta un modello di turbolenza basato sulle equazioni RANS e sulla viscosità turbolenta, è necessario risolvere un set di equazioni per determinare questa viscosità. Una definizione di viscosità turbolenta si basa sulla velocità  $U$  e sulla lunghezza  $l$  caratterizzanti la scala maggiore degli eddies turbolenti.  $C_v$  è una costante di proporzionalità

$$\nu_t = C_v ul. \quad (2.33)$$

Ora, secondo la teoria di Kolmogorov della turbolenza omogenea ed isotropa,

$$U = k^{0.5} \quad (2.34)$$

$$l = \frac{k^{1.5}}{\epsilon}. \quad (2.35)$$

Da ciò è possibile ricavare l'espressione di viscosità turbolenta mostrata dall'equazione (2.36).

Ora, esistono numerosi metodi che sono in grado di fornire valori di viscosità turbolenta. Questi metodi sono caratterizzati dal numero di equazioni di trasporto aggiuntive che sono richieste per la chiusura, ovvero le equazioni necessarie per determinare la velocità e la lunghezza di scala caratterizzanti la turbolenza locale. Fra questi modelli, i modelli a due equazioni sono i più utilizzati negli algoritmi di simulazione utilizzati per applicazioni industriali. All'interno dei modelli a due equazioni, il modello  $k - \epsilon$  Standard rappresenta un compromesso fra generalità ed economia della simulazione.

Il modello  $k - \epsilon$  Standard descrive la turbolenza usando due variabili: l'energia cinetica turbolenta  $k$  e la velocità di dissipazione dell'energia cinetica turbolenta  $\epsilon$ . Per ciascuna di queste proprietà viene implementata un'equazione di trasporto. In conclusione, all'interno di questo modello è necessario risolvere il set di equazioni dato da

$$\nu_t = C_\mu \frac{k^2}{\epsilon} \quad (2.36)$$

$$\frac{\partial k}{\partial t} + \langle \vec{U} \rangle \cdot \nabla k = \nu_t [(\nabla \langle \vec{U} \rangle + (\nabla \langle \vec{U} \rangle)^T) \nabla \langle \vec{U} \rangle] - \epsilon + \nabla \cdot \left[ \left( \nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right] \quad (2.37)$$

$$\frac{\partial \epsilon}{\partial t} + \langle \vec{U} \rangle \cdot \nabla \epsilon = C_{\epsilon 1} \nu_t \frac{\epsilon}{k} [(\nabla \langle \vec{U} \rangle + (\nabla \langle \vec{U} \rangle)^T) \nabla \langle \vec{U} \rangle] - C_{\epsilon 2} \frac{\epsilon^2}{k} + \nabla \cdot \left[ \left( \nu + \frac{\nu_t}{\sigma_\epsilon} \right) \nabla \epsilon \right]. \quad (2.38)$$

Nel modello  $k - \epsilon$  sono presenti cinque coefficienti, che permettono la chiusura, assunti universali e quindi costanti. Questi coefficienti sono riportati in Tabella 2.1.

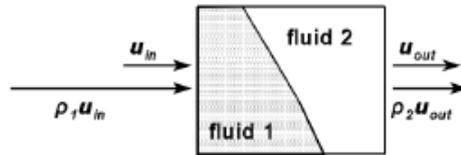
| Costante          | Valore |
|-------------------|--------|
| $C_\mu$           | 0.09   |
| $C_{\epsilon 1}$  | 1.44   |
| $C_{\epsilon 2}$  | 1.92   |
| $\sigma_k$        | 1.00   |
| $\sigma_\epsilon$ | 1.30   |

**Tabella 2.1:** Coefficienti di chiusura Modello  $k - \epsilon$  Standard.

## 2.4 Metodi per il tracciamento dell'interfaccia

Come menzionato nella sezione 2.2, all'interno del campo delle simulazioni multifase, emerge il problema della discontinuità della quantità di moto attraverso la superficie

libera fra le due fasi, osservabile in Figura 2.7. Infatti,  $\rho\vec{U}$  non è continua in tutto il dominio fluido a causa del salto di densità all'interfaccia fra le due fasi. Come conseguenza di questa discontinuità, l'equazione (2.17) non può essere impiegata perché la funzione  $\rho\vec{U}$  dovrebbe essere continua e differenziabile su tutto il dominio. Dunque è necessario riscrivere l'equazione di bilancio della quantità di moto in modo che essa sia impiegabile anche nel campo delle simulazioni multifase. Per



**Figura 2.7:** Continuità delle velocità e discontinuità della quantità di moto nel flusso bifasico.

questa ragione sono stati proposti metodi, in grado di superare questa problematica, adatti ad analizzare e calcolare numericamente i flussi attraverso la superficie libera. In letteratura esistono due categorie di metodi per il calcolo dei flussi attraverso la superficie libera:

- metodi di superficie: in questi metodi la superficie libera viene tracciata in modo esplicito tramite l'uso di speciali particelle traccianti (marker particles) oppure forzando la griglia computazionale a rimanere attaccata alla superficie libera e a muoversi con essa. Quando un metodo di superficie è usato allora la corrispondente tecnica numerica viene definita "surface fitting"
- metodi di volume: in questa seconda categoria, la superficie libera viene tracciata usufruendo di particelle prive di massa o di una funzione indicatrice. Quando un metodo di volume è usato allora la corrispondente tecnica numerica viene definita "surface capturing".

All'interno di questa tesi verranno solo menzionati e brevemente illustrati i metodi esistenti e ci si focalizzerà sul Volume of Fluid Method (VOF), il quale è stato sfruttato all'interno delle simulazioni del riempimento realizzate in questo lavoro di tesi.

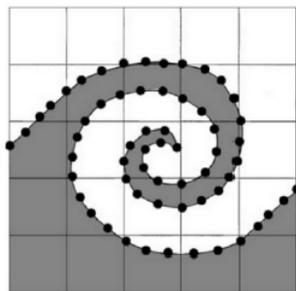
### 2.4.1 Metodi di superficie

Analizzando i metodi di superficie (surface fitting), come è stato già spiegato, la superficie libera viene, in primo luogo, seguita nella sua evoluzione tramite alcuni innumerevoli punti (marker points) posti sull'interfaccia, e, in secondo luogo

viene utilizzato un metodo di interpolazione al fine di individuare e rappresentare la superficie fra questi punti. È possibile distinguere diversi modi principali di marcatura delle superficie libere.

### Interface particles

Un set di particelle prive di massa vengono sparse su tutta la superficie libera e ogni particella viene trascinata per moto convettivo dalla velocità del fluido presente nella zona della particella, ovvero in maniera lagrangiana. Questo metodo non è molto accurato quando la dispersione di particelle non è fitta. Infatti, più è la distanza fra le particelle e più si compie un errore nella ricostruzione dell'interfaccia tramite interpolazione. Inoltre, è richiesta una grande memoria per immagazzinare la posizione in ogni istante temporale di tutte le particelle marker. All'interno della Figura 2.8 è possibile osservare la distribuzione all'interfaccia delle particelle prive di massa.



**Figura 2.8:** Interface particles method.

### Height functions

La superficie libera viene identificata definendo la sua distanza da una linea di riferimento come funzione della posizione lungo la stessa linea di riferimento. Viene così definita la funzione altezza  $\eta(x, t)$  la cui evoluzione nello spazio e nel tempo, all'interno di un modello bidimensionale, è governata dalla seguente equazione

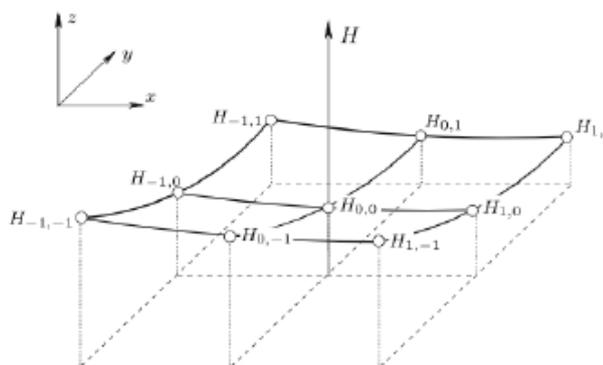
$$\frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} = v \quad (2.39)$$

dove  $(u, v)$  sono le componenti del vettore velocità lungo  $(x, y)$ .

Tramite la condizione espressa dalla (2.39) si impone alla superficie di muoversi con fluido. Si noti come l'equazione abbia un ruolo cruciale all'interno della teoria delle onde. Questa condizione al contorno cinematica è adatta solo per problemi bi-dimensionali e ha severe limitazioni in quanto non riesce a descrivere bene

tutti i tipi di forma che può assumere una superficie libera. Infatti, la soluzione dell'equazione differenziale è una funzione che per definizione è una relazione tra due insiemi, chiamati dominio e codominio della funzione, che associa a ogni elemento del dominio uno e un solo elemento del codominio. Da quest'ultima affermazione appare evidente come superfici libere delimitanti bolle e/o gocce, oggetti molto presenti all'interno di apparecchiature chimiche e alimentari, non possano essere rappresentate.

All'interno della Figura 2.9 è possibile osservare la funzione altezza ( $H$ ) per alcuni punti costituenti l'interfaccia.



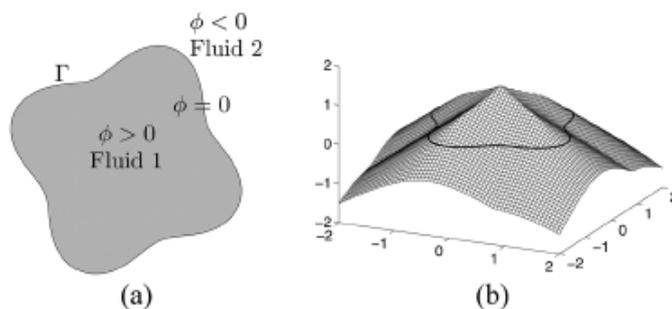
**Figura 2.9:** Height functions method.

### Level-set methods

Una funzione continua scalare  $\phi(x, y, z, t)$ , definita funzione level-set, viene introdotta in tutto il dominio fluido. Il valore della funzione level-set in ogni punto del dominio è definito come la minima distanza fra quel preciso punto e l'interfaccia. Da ciò emerge come la superficie libera sia collocata per definizione in quei punti dove la funzione ha valore zero. Al fine di distinguere quale dei due fluidi si trovi da una parte e quale dall'altra parte dell'interfaccia, si assegnerà, in modo arbitrario, un segno negativo alla funzione scalare di uno dei due fluidi, come è mostrato in Figura 2.10. In questo modo il fluido ha una proprietà scalare, la funzione level-set, che si trasporta mediante un'equazione scalare convettiva. Con questo metodo è possibile avere evidenti problemi numerici emergenti ad esempio dalla fusione di due bolle, dove si ha l'intersezione di due regioni di superficie libera ( $\phi = 0$ ).

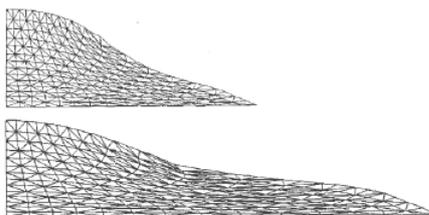
### Surface fitted methods

Questi metodi comprendono tutte quelle tecniche nelle quali la griglia computazionale è attaccata all'interfaccia. Un esempio di ciò è visibile in Figura 2.11.



**Figura 2.10:** Level set method.

Questi metodi sono limitati alle superfici libere non soggette a grandi deformazioni poiché ciò provocherebbe una significativa deformazione e distorsione della griglia. Essendo il metodo molto sensibile alla griglia di calcolo, allora si potrebbero avere errori numerici che altererebbero il significato fisico del risultato della simulazione.



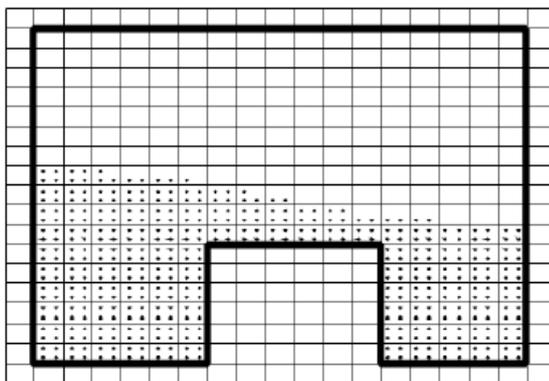
**Figura 2.11:** Surface fitted method.

## 2.4.2 Metodi di volume

### Marker particles

Questa tecnica prevede la presenza di particelle marcatrici non solo all'interfaccia, come accadeva nel metodo "interface particles", ma su tutta la regione occupata dal fluido, come è visibile in Figura 2.12. Da ciò emerge la differenza principale fra le dizioni "metodi di volume" e "metodi di superficie". Ogni particella marcatrice situata in un certo punto del dominio fluido viene fatta muovere con la velocità che possiede il fluido in quel preciso punto. Il moto di queste particelle è, dunque, definito in modo lagrangiano, proprio come accadeva anche nel caso della tecnica "interface particles". La superficie libera viene definita osservando quelle celle della griglia di calcolo contenenti particelle marcatrici che hanno, però, almeno una cella confinante in cui non sono presenti particelle. Questo metodo può trattare fenomeni complessi e può essere usato anche in calcoli il cui dominio è

tridimensionale. Richiede, inoltre, molta potenza di calcolo e grande quantità di memoria per immagazzinare le informazioni necessarie a seguire il moto di un'enorme quantitativo di particelle.



**Figura 2.12:** Marker particles method.

### Volume fractions

Questi metodi comprendono molte tecniche, fra le quali la principale e più usata risulta essere il metodo VOF. Questo metodo è stato sfruttato all'interno del lavoro di tesi al fine di simulare il riempimento ed è stato introdotto da Hirt e Nichols nel 1981 [1].

Un fattore che accomuna tutti i metodi “volume fractions” è l'utilizzo di una funzione scalare indicatrice  $\alpha(\vec{x}, t)$  che rappresenta la frazione di volume, di una cella, occupata da una delle fasi costituenti il sistema. Nel caso di sistema composto da due fasi, come nel caso di questa tesi, allora sarà necessario l'impiego di una sola funzione indicatrice ed essa indicherà la presenza o l'assenza di una delle due fasi. In aggiunta, risulta ovvio che nel caso di sistema con un più di due fasi allora saranno necessarie più funzioni indicatrici. All'interno di questa tesi, le due fasi considerate risultano essere il liquido e il gas di riempimento. Da ciò, si può esprimere la funzione indicatrice come

$$\alpha = \frac{V_{fluid}}{V_{cell}} \quad (2.40)$$

$$\alpha = \begin{cases} 1 & \text{se } V_{fluid} = V_{cell}, \\ 0 & \text{se } V_{fluid} = 0, \\ 0 < \alpha < 1 & \text{se } V_{fluid} < V_{cell}. \end{cases} \quad (2.41)$$

Come si può vedere in Figura 2.13, l'interfaccia si colloca nelle celle in cui  $\alpha$  è compreso fra 0 e 1. Il principale vantaggio di questo metodo è la richiesta moderata di potenza di calcolo rispetto alla tecnica che utilizza le particelle marcatrici. Infatti, è necessario immagazzinare in memoria una sola variabile, la funzione indicatrice, e non più i dati relativi ad un grande numero di particelle. Ovviamente, ciò comporta una peggior risoluzione e accuratezza dei metodi "volume fractions". In queste tecniche sono presenti due algoritmi: il primo algoritmo, chiamato algoritmo di advezione, calcola accuratamente l'evoluzione della funzione indicatrice; il secondo algoritmo, invece, chiamato algoritmo di ricostruzione, ricostruisce e delinea la superficie libera una volta che il valore della funzione indicatrice è noto in ogni cella.

Negli anni, molti algoritmi di advezione e ricostruzione sono stati proposti e

|      |      |      |      |
|------|------|------|------|
| 0.09 | 0.22 | 0.00 | 0.00 |
| 0.96 | 1.00 | 0.64 | 0.68 |
| 1.00 | 1.00 | 1.00 | 1.00 |

**Figura 2.13:** Volume fractions method.

sviluppati al fine di migliorare sempre più la risoluzione e l'accuratezza della posizione dell'interfaccia in questi tipi di metodo.

In questa tesi, il focus principale viene posto solo sull'algoritmo VOF proposto nel lavoro originale di Hirt e Nichols [1].

### 2.4.3 Volume of Fluid Method

Il metodo VOF è probabilmente il metodo più usato per analizzare flussi interfacciali. Esso è presente all'interno del solver opensource InterFOAM, che consente di descrivere flussi multifase. Esso è un solver disponibile fra quelli standard della libreria CFD software OpenFOAM.

All'interno del metodo VOF, il trasporto della funzione indicatrice è governato dalla seguente relazione:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{U} \alpha) = 0 \quad (2.42)$$

Per i fluidi incomprimibili, la conservazione della massa è equivalente alla conservazione del volume e quindi, l'equazione (2.42) esprime la conservazione della frazione

volumica.

Il modo con il quale il flusso di  $\alpha$  viene valutato da una cella (donatore) ad una cella confinante (accettore), ovvero l'algoritmo di avezione, insieme alla tecnica di ricostruzione utilizzata per definire l'interfaccia, definiscono lo schema "donatore-accettore" ("donor-acceptor scheme"), il quale viene illustrato sia nell'articolo originale di Hirt e Nichols [1] e sia nel lavoro di Ubbink [2], propositore dell'algoritmo InterFOAM per i flussi multifase.

Il metodo VOF è stato introdotto per ottenere la continuità della funzione quantità di moto  $\rho\vec{U}$  attraverso l'interfaccia e modellare due fluidi immiscibili come se fossero un unico fluido continuo. Per ottenere ciò, non è possibile usare la definizione di  $\alpha$  come funzione a gradino come mostrato nell'equazione (2.41): si ha il bisogno di definire una regione di transizione, localizzata fra i due fluidi, di spessore finito  $\delta$ , all'interno della quale la funzione indicatrice possa variare in modo continuo e regolare fra 0 e 1. Questa regione è solamente un artificio matematico e non esiste fisicamente, ma è una buona approssimazione del comportamento reale all'interfaccia. Il requisito di variazione continua e regolare della funzione indicatrice non è data dalla funzione densità all'interno della (2.8), ma è data dall'equazione usata per valutare la curvatura dell'interfaccia (2.20), espressa dalla teoria delle tensioni superficiali.

Da questo modello emerge come la funzione indicatrice necessiti di essere di classe  $C^2$  su tutto il dominio. Come conseguenza di ciò, è possibile definire la funzione indicatrice di classe  $C^2$  che soddisfa i seguenti requisiti:

$$\alpha(\vec{x}, t) = \begin{cases} 1 & \text{se } (\vec{x}, t) \text{ è all'interno del fluido 1 (acqua)} \\ 0 & \text{se } (\vec{x}, t) \text{ è all'interno del fluido 2 (aria),} \\ 0 < \alpha_\delta < 1 & \text{se } (\vec{x}, t) \text{ è all'interno della regione di transizione } \delta. \end{cases} \quad (2.43)$$

La sua evoluzione nello spazio e nel tempo è descritta dalla (2.42).

Ora, grazie ai concetti introdotti, è possibile utilizzare una singola equazione di quantità di moto per il sistema composto da due fluidi (nel caso di questa tesi acqua e anidride carbonica), definendo due nuove funzioni continue per densità e viscosità come segue:

$$\begin{cases} \rho = \rho_w \alpha + \rho_{CO_2}(1 - \alpha) \\ \mu = \mu_w \alpha + \mu_{CO_2}(1 - \alpha). \end{cases} \quad (2.44)$$

Attraverso le equazioni (2.42) (2.43) e (2.44) si completa il modello di tracciamento dell'interfaccia.

Le equazioni presenti in questo modello presentano alcune peculiarità e problemi e sono necessari alcuni accorgimenti al fine di giungere alla loro risoluzione.

## 2.5 Modello matematico adottato nel solver InterFOAM

Riassumendo, nel metodo VOF convenzionale l'equazione di trasporto della funzione indicatrice viene risolta simultaneamente con l'equazione di continuità e di quantità di moto

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0 \quad (2.45)$$

$$\frac{\partial(\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \vec{U}) = -\nabla p + \nabla \cdot (2\mu \vec{S} - 2\mu(\nabla \cdot \vec{U})\vec{I}/3) + \rho \vec{f}_b \quad (2.46)$$

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{U} \alpha) = 0 \quad (2.47)$$

Due fluidi immiscibili sono considerati come un unico effettivo fluido all'interno del dominio. Le proprietà fisiche di esso sono calcolate come media pesata basata sulla distribuzione della frazione volumica di liquido

$$\begin{cases} \rho = \rho_l \alpha + \rho_g(1 - \alpha) \\ \mu = \mu_l \alpha + \mu_g(1 - \alpha) \end{cases} \quad (2.48)$$

dove  $\rho_l$  e  $\rho_g$  sono rispettivamente la densità del liquido e la densità del gas.

Uno dei problemi critici nelle simulazioni dei flussi di superficie libera con il metodo VOF è la conservazione della funzione indicatrice, ovvero della frazione di fase. Ciò si manifesta soprattutto nel caso in cui si vogliono simulare fluidi con alti rapporti di densità, dove piccoli errori nel calcolo della frazione volumica possono provocare errori significativi nel calcolo delle proprietà fisiche.

Un calcolo accurato della distribuzione della frazione di fase è cruciale affinché si abbia una valutazione corretta della curvatura di superficie, la quale è richiesta per la determinazione della tensione superficiale e il corrispondente gradiente di pressione attraverso la superficie. La regione d'interfaccia fra la due fasi è tipicamente spalmata su più celle della griglia e quindi è altamente sensibile alla risoluzione delle griglia.

In campo computazionale, non è semplice assicurare la limitatezza e la conservazione della frazione di fase. Vari tentativi sono stati fatti al fine di superare queste difficoltà. Inoltre, la definizione di velocità su cui si basa l'avanzamento della superficie libera, come una singola velocità condivisa fra le due fasi, risulta essere fallace: nessuna conclusione può essere tratta in merito alla misura con cui la velocità di ogni fase contribuisce alla velocità dell'effettivo fluido.

All'interno del solver InterFOAM il modello matematico convenzionale del metodo Volume of Fluid viene modificato. In questo modello un termine aggiuntivo

convettivo, originato dalla modellazione della velocità in termini di media pesata delle corrispondenti velocità liquide e gas, è introdotto nell'equazione di trasporto della frazione di fase, fornendo una più nitida risoluzione dell'interfaccia. Il modello fa uso del modello a due fluidi Euleriano per flussi bifasici, dove le equazioni della frazione di fase sono risolte separatamente per ciascuna fase individuale. Dunque, le equazioni per ciascuna delle frazioni di fase possono essere espresse come

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{U}_l \alpha) = 0 \quad (2.49)$$

$$\frac{\partial (1 - \alpha)}{\partial t} + \nabla \cdot (\vec{U}_g (1 - \alpha)) = 0 \quad (2.50)$$

dove i pedici  $l$  e  $g$  indicano rispettivamente la fase liquida e gassosa. Si definisca inoltre il vettore velocità relativa, chiamato anche velocità di compressione, come

$$\vec{U}_r = \vec{U}_l - \vec{U}_g. \quad (2.51)$$

Ora, assumendo che il contributo delle velocità del gas e del liquido all'evoluzione della superficie libera sia proporzionale alla corrispondente frazione di fase e definendo la velocità dell'effettivo fluido in un modello VOF come media pesata, è possibile scrivere

$$\vec{U} = \alpha \vec{U}_l + (1 - \alpha) \vec{U}_g. \quad (2.52)$$

Dalla definizione di velocità relativa si ha che

$$\vec{U}_g = \vec{U}_l - \vec{U}_r. \quad (2.53)$$

Ora, sostituendo la (2.53) all'interno della (2.52) si ottiene che

$$\vec{U} = \alpha \vec{U}_l + (1 - \alpha) \vec{U}_l - (1 - \alpha) \vec{U}_r \quad (2.54)$$

da cui

$$\vec{U} = \vec{U}_l - (1 - \alpha) \vec{U}_r \quad (2.55)$$

e quindi

$$\vec{U}_l = \vec{U} + (1 - \alpha) \vec{U}_r. \quad (2.56)$$

Infine, sostituendo la (2.56) all'interno della (2.49) si ottiene

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{U} \alpha) + \nabla \cdot (\vec{U}_r \alpha (1 - \alpha)) = 0. \quad (2.57)$$

L'equazione (2.57) ha una difficile interpretazione dal punto di vista fisico, ma è adatta a comprimere l'interfaccia nella formulazione discreta, specialmente quando l'interfaccia non è abbastanza netta. L'equazione di governo della frazione volumica

(2.57) contiene un termine aggiuntivo convettivo, denominato termine di compressione in quanto il suo ruolo è comprimere la superficie libera in modo da averne una più netta. Inoltre, è bene precisare che il termine "compressione" viene usato in questa trattazione solo come denotazione e non è legato in alcun modo a fluidi comprimibili.

Confrontando le equazioni (2.57) e (2.47) emerge come il termine aggiuntivo appaia come un contributo artificiale alla convezione della frazione di fase. Siccome la derivazione dell'equazione (2.57) si basa sulla definizione di velocità relativa allora si ha una forte assonanza tra il modello VOF e ed classico modello a due fasi Euleriano. Il termine aggiuntivo convettivo contribuisce in maniera significativa all'ottenimento di un'alta risoluzione d'interfaccia. Questo termine è attivo solamente nei limiti della regione d'interfaccia ed è nullo ad entrambi i limiti della frazione di fase. Quindi, esso non influisce sulla soluzione fuori da questa regione. In aggiunta, se la superficie libera è definita in senso teorico come avente uno spessore infinitamente piccolo, la velocità relativa  $\vec{U}_r$  si annulla e l'espressione (2.57) si riduce alla forma convenzionale (2.47).

Oltre a riflettere una corretta fisica dei flussi, il principale vantaggio di questa formulazione è nella possibilità di catturare la regione d'interfaccia in maniera netta. La diffusione numerica, inevitabilmente introdotta attraverso la discretizzazione dei termini convettivi, illustrata in seguito, può esser controllata e minimizzata attraverso la discretizzazione del termine di compressione, consentendo così una netta risoluzione d'interfaccia. I dettagli su ciò verranno discussi in seguito.

L'equazione di trasporto della quantità di moto è modificata al fine di tener conto degli effetti della tensione superficiale. La tensione superficiale all'interfaccia liquido-gas genera un aggiuntivo gradiente di pressione che produce una forza, la quale è valutata per unità di volume attraverso il modello di forza di superficie continua (CSF), come già espresso precedentemente. Si ricorda, dunque, che

$$\vec{f}_\gamma = \gamma \kappa \nabla \alpha \quad (2.58)$$

dove  $\kappa$  è la curvatura media della superficie libera, determinata dall'espressione

$$\kappa = -\nabla \cdot \left( \frac{\nabla \alpha}{|\nabla \alpha|} \right). \quad (2.59)$$

L'equazione (2.58) è solo valida per i casi con tensione superficiale costante, come è considerata in questa discussione. Nel caso di tensione superficiale variabile, ad esempio causata da una distribuzione non uniforme della temperatura, si incontrano gradienti di tensione superficiale i quali generano un aggiuntivo sforzo di taglio all'interfaccia, il quale deve essere preso in considerazione.

Entrambi i fluidi sono considerati newtoniani ed incomprimibili e il tensore velocità di deformazione viene correlato linearmente al tensore degli sforzi deviatorico, il

quale è decomposto in una più conveniente forma per la discretizzazione

$$\nabla \cdot \mathbf{D} = \nabla \cdot (\mu(\nabla \vec{U} + (\nabla \vec{U})^T)) = \nabla \cdot (\mu \nabla \vec{U}) + (\nabla \vec{U}) \cdot \nabla \mu. \quad (2.60)$$

E' pratica comune, all'interno dei modelli sfruttanti la teoria del VOF, definire una pressione modificata, al fine di semplificare le condizioni al contorno

$$p_d = p - \rho \vec{g} \cdot \vec{x} \quad (2.61)$$

dove  $\vec{x}$  è il vettore posizione. All' interno dell'equazione di trasporto di quantità di moto, il gradiente di pressione viene espresso tramite l'utilizzo della pressione modificata. Si ottiene dunque

$$\frac{\partial(\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \vec{U}) - \nabla \cdot (\mu \nabla \vec{U}) - (\nabla \vec{U}) \cdot \nabla \mu = -\nabla p_d - \vec{g} \cdot \vec{x} \nabla \rho - \gamma \nabla \cdot \left( \frac{\nabla \alpha}{|\nabla \alpha|} \right) \nabla \alpha \quad (2.62)$$

Le forze di volume causate dal gradiente di pressione e dalla gravità vengono tenute conto tramite i primi due termine del lato destro dell'equazione (2.62). In conclusione, il modello matematico è dato dall'equazione di continuità (2.45) , dall'equazione di trasporto della frazione di fase (2.57) e dall'equazione di trasporto della quantità di moto (2.62). Il modello viene chiuso fornendo un'espressione appropriata per la velocità di compressione  $\vec{U}_r$ . Essa è fornita dal lavoro di Weller [3], il quale suggerisce di calcolare la velocità di compressione come

$$\vec{U}_r = C_\alpha |\vec{U}| \vec{n} \quad (2.63)$$

L'intensità della compressione dell'interfaccia è governata dalla costante  $C_\alpha$ , la quale può avere un valore superiore o uguale a 0. Inserendo un valore pari a zero si ha compressione nulla d'interfaccia, mentre adottando un valore pari a 1 si ha una compressione conservativa, come è stato adottato all'interno di questa tesi. Infine, adottando un valore maggiore di 1 si ha una grande compressione d'interfaccia.

Al fine di assicurare che la velocità  $\vec{U}_r$  non pregiudichi la soluzione in alcun modo, essa deve agire solo in direzione perpendicolare all'interfaccia. In aggiunta, osservando la (2.62) è evidente che solo i valori di  $\vec{U}_r$  sulle facce della griglia verranno usati, in accordo con la discretizzazione del termine convettivo, come verrà ampiamente discusso in seguito.

### 2.5.1 Equazioni del modello

Per concludere il capitolo, vengono riportate le equazioni risolte per il risolvere il problema della simulazione del riempimento.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0 \quad (2.64)$$

$$\frac{\partial(\rho\vec{U})}{\partial t} + \nabla \cdot (\rho\vec{U}\vec{U}) - \nabla \cdot (\mu\nabla\vec{U}) - (\nabla\vec{U}) \cdot \nabla\mu = -\nabla p_d - \vec{g} \cdot \vec{x} \nabla\rho - \gamma \nabla \cdot \left( \frac{\nabla\alpha}{|\nabla\alpha|} \right) \nabla\alpha \quad (2.65)$$

$$\frac{\partial\alpha}{\partial t} + \nabla \cdot (\vec{U}\alpha) + \nabla \cdot (\vec{U}_r\alpha(1-\alpha)) = 0. \quad (2.66)$$

$$\vec{U}_r = C_\alpha |\vec{U}| \nabla\alpha \quad (2.67)$$

$$\rho = \rho_l\alpha + \rho_g(1-\alpha) \quad (2.68)$$

$$\mu = \mu_l\alpha + \mu_g(1-\alpha) \quad (2.69)$$

## Capitolo 3

# Soluzione numerica delle equazioni del modello

### 3.1 Metodo ai volumi finiti

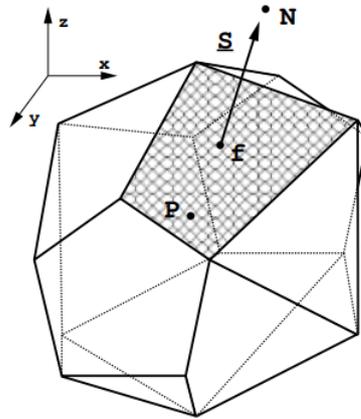
La costruzione di un metodo numerico per ottenere la soluzione delle equazioni di trasporto presentate nel precedente capitolo richiede che il dominio fluido venga rappresentato come una griglia di volumi discreti (celle) dove ogni termine delle equazioni differenziali alle derivate parziali viene valutato. Tutto ciò produce un sistema di equazioni algebriche accoppiate che necessita una soluzione simultanea data da un algoritmo risolutore. Infatti, l'obiettivo di ogni procedura di discretizzazione è trasformare una o più equazioni differenziali alle derivate parziali in un sistema di equazioni algebriche. La soluzione di questo sistema produce un insieme di valori che corrispondono alla soluzione delle equazioni in punti precisi nello spazio e nel tempo. La procedura di discretizzazione può essere divisa in due parti: la discretizzazione del dominio fluido e la discretizzazione delle equazioni, che a sua volta si divide in spaziale e temporale.

Il metodo di discretizzazione utilizzato all'interno di questa tesi è il noto Metodo ai Volumi Finiti. Questo metodo prevede la suddivisione del dominio fluido in un numero finito di piccoli e non sovrapposti volumetti di controllo. Le equazioni di trasporto, dunque, vengono integrate su ciascun volume di controllo approssimando la variazione delle proprietà fluide tra i punti della griglia con profili a tratti. Questa integrazione assicura che la legge di bilancio, utilizzata per la derivazione delle equazioni di trasporto, sia soddisfatta localmente per ciascun volume di controllo. In conclusione, la conservazione globale delle proprietà fluide è quindi automaticamente preservata. Queste approssimazioni a tratti, anche note come schemi di differenziazione, sono costruiti al fine di supportare aspetti fisici del comportamento fluido come convezione o diffusione.

## 3.2 Discretizzazione del dominio fluido

In questa fase si considera la zona spaziale di simulazione e si produce una griglia (o mesh) sulla quale le equazioni di trasporto vengono risolte. Nel metodo dei volumi finiti il dominio viene suddiviso in volumi di controllo. Essi non si devono sovrapporre e la loro somma volumica deve essere uguale al volume della zona fluida che si vuole simulare.

Una tipica struttura di volume di controllo è presentata in Figura 3.1. Il punto  $P$



**Figura 3.1:** Esempio di volume di controllo in cui suddiviso il dominio fluido

rappresenta il centro della cella. Il volume di controllo è circondato da una serie di facce piane e ogni faccia è condivisa con un solo volume di controllo confinante. La topologia di questi volumetti non è importante: di solito sono poliedri. Le facce della cella possono essere divise in due sottogruppi: le facce interne (tra due volumi di controllo) o esterne, che coincidono con la superficie geometrica che racchiude il dominio fluido. Il vettore superficie  $\vec{S}_f$  viene costruito per ogni faccia ed è rivolto verso l'esterno della cella. Esso inoltre ha magnitudo pari alla superficie della faccia e direzione ortogonale alla faccia stessa. Le celle vengono contrassegnate da un'etichetta (label). La cella contrassegnata con l'etichetta minore viene definita proprietaria della faccia (owner). La cella che condivide la stessa faccia sarà denominata cella confinante (neighbour). Nella (Figure 4.1) il centro della cella proprietaria è il punto  $P$ , mentre il centro della cella confinante è il punto  $N$ . Per semplicità, all'interno di questo lavoro tutte le facce del volume di

controllo saranno contrassegnate con la lettera  $f$ , la quale rappresenta inoltre il punto al centro della faccia.

### 3.3 Discretizzazione delle equazioni

La forma standard di una generica equazione di trasporto per una quantità scalare  $\phi$  è definita come

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \vec{U} \phi) - \nabla \cdot (\rho \Gamma_\phi \nabla \phi) = S_\phi(\phi) \quad (3.1)$$

Il primo termine è un termine di accumulo o variazione temporale, il secondo termine rappresenta il trasporto convettivo, il terzo termine è un termine diffusivo. Infine, si ha un termine di generazione.

Il metodo dei Volumi Finiti richiede che l'equazione (3.1) sia soddisfatta sul volume di controllo generico  $V_p$ , nel quale centro è presente il punto  $P$ , nella forma integrale

$$\int_t^{t+\Delta t} \left( \frac{\partial}{\partial t} \int_{V_p} \rho \phi dV + \int_{V_p} \nabla \cdot (\rho \vec{U} \phi) dV - \int_{V_p} \nabla \cdot (\rho \Gamma_\phi \nabla \phi) dV \right) dt = \int_t^{t+\Delta t} \left( \int_{V_p} S_\phi(\phi) dV \right) dt \quad (3.2)$$

L'accuratezza del metodo di discretizzazione dipende dalla variazione assunta della funzione  $\phi = \phi(\vec{x}, t)$  attorno al punto  $P$ . Si assuma che

$$\phi(\vec{x}) = \phi_P + (\vec{x} - \vec{x}_P) \cdot (\nabla \phi)_P \quad (3.3)$$

dove

$$\phi_P = \phi(\vec{x}_P) \quad (3.4)$$

É possibile, dunque, scrivere che

$$\begin{aligned} \int_{V_P} \phi(\vec{x}) dV &= \int_{V_P} (\phi_P + (\vec{x} - \vec{x}_P) \cdot (\nabla \phi)_P) dV = \\ \phi_P \int_{V_P} dV + (\int_{V_P} (\vec{x} - \vec{x}_P) dV) \cdot (\nabla \phi)_P &= \phi_P V_P \end{aligned} \quad (3.5)$$

dove  $V_P$  è il volume della cella. Nell'equazione (3.5), il secondo integrale risulta essere zero perché il punto  $P$  è il centroide del volume di controllo.

Ora, al fine di ottenere la discretizzazione dei termini spaziali è utile ricordare che

$$\int_V \nabla \cdot \vec{a} dV = \oint_{\partial V} \vec{a} \cdot d\vec{S} \quad (3.6)$$

$$\int_V \nabla \phi dV = \oint_{\partial V} \phi d\vec{S} \quad (3.7)$$

$$\int_V \nabla \vec{a} dV = \oint_{\partial V} \vec{a} \otimes d\vec{S} \quad (3.8)$$

dove  $\partial V$  è la superficie chiusa costituente il bordo di  $V$ , mentre  $dS$  rappresenta un elemento infinitesimo di superficie.

Sapendo che il volume di controllo è circondato da una serie di facce piane, l'equazione (3.6) può essere trasformata in una somma di integrali su tutte le facce

$$\int_{V_P} \nabla \cdot \vec{a} dV = \oint_{\partial V_P} \vec{a} \cdot d\vec{S} = \sum_f \left( \int_f \vec{a} \cdot d\vec{S} \right) \quad (3.9)$$

assumendo che

$$\vec{a} = \vec{a}_f + (\vec{x} - \vec{x}_P) \cdot (\nabla \vec{a})_P \quad (3.10)$$

allora

$$\int_f \vec{a} d\vec{S} = \left( \int_f d\vec{S} \right) \cdot \vec{a}_f + \left( \int_f (\vec{x} - \vec{x}_P) d\vec{S} \right) \cdot (\nabla \vec{a})_f = \vec{S} \cdot \vec{a}_f \quad (3.11)$$

Combinando le equazioni (3.5), (3.9) e (3.11) si può ottenere

$$(\nabla \cdot \vec{a})_{V_P} = \sum_f \vec{S} \cdot \vec{a}_f \quad (3.12)$$

Nella Figura 3.1, il vettore  $\vec{S}_f$  ha direzione uscente da  $P$  solo se  $f$  è posseduta da  $P$ . Per le celle confinanti,  $\vec{S}_f$  punta verso l'interno. Per tenere conto di ciò nella somma dell'equazione (3.12), la somma su tutte le facce viene divisa come

$$\sum_f \vec{S} \cdot \vec{a}_f = \sum_{owner} \vec{S}_f \cdot \vec{a}_f - \sum_{neighbour} \vec{S}_f \cdot \vec{a}_f \quad (3.13)$$

### 3.3.1 Discretizzazione del termine di convezione

Per quanto riguarda il termine di convezione, la sua discretizzazione viene ottenuta utilizzando l'equazione (3.12)

$$\int_{V_P} \nabla \cdot (\rho \vec{U} \phi) dV = \sum_f \vec{S} \cdot (\rho \vec{U} \phi)_f = \sum_f \vec{S} \cdot (\rho \vec{U})_f \phi_f = \sum_f F \phi_f \quad (3.14)$$

dove  $F$  nell'equazione (3.14) rappresenta il flusso massico attraverso la faccia

$$F = \vec{S} \cdot (\rho \vec{U})_f \quad (3.15)$$

Il calcolo di questi flussi verrà discusso in seguito. Per ora si può ritenere che il flusso massico venga calcolato da un'interpolazione dei valori di  $\rho$  e  $\vec{U}$ . Si noti come

nell'equazione (3.14) sia richiesto il valore della variabile  $\phi$  al centro della faccia. Il ruolo della discretizzazione dei termini convettivi è ottenere il valore al centro della faccia partendo dai valori al centro delle celle. Nella, pratica si considerano solo i valori al centro delle celle confinanti con la faccia che si vuole discretizzare. Al fine di ottenere il valore di  $\phi_f$  è possibile utilizzare uno schema Central Differencing (CD). Esso è uno schema di secondo ordine e assume una variazione lineare di  $\phi$  fra  $P$  ed  $N$ , come visibile in Figura 3.2

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N. \quad (3.16)$$

Il fattore d'interpolazione  $f_x$  si definisce come un rapporto fra distanze

$$f_x = \frac{f\bar{N}}{P\bar{N}}. \quad (3.17)$$

Un'alternativa allo schema CD è rappresentata dall' Upwind Differencing Schemes (UD), dove il valore al centro della faccia è determinato in base alla direzione del flusso.

$$\phi_f = \begin{cases} \phi_f = \phi_P, & \text{per } F > 0, \\ \phi_f = \phi_N, & \text{per } F < 0. \end{cases} \quad (3.18)$$

Questo schema è meno accurato in quanto introduce diffusione numerica all'interno del calcolo, in quanto è uno schema del primo ordine in cui l'errore di discretizzazione è proporzionale alla dimensione della cella.

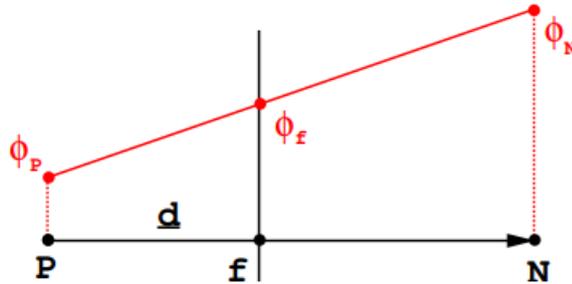


Figura 3.2: Interpolazione nel CDS

### 3.3.2 Discretizzazione del termine di diffusione

Per quanto riguarda il termine di diffusione, sfruttando l'equazione (3.12) e la variazione lineare di  $\phi$  si può scrivere che

$$\int_{V_P} \nabla \cdot (\rho \Gamma_\phi \nabla \phi) dV = \sum_f \vec{S} \cdot (\rho \Gamma_\phi \nabla \phi)_f = \sum_f (\rho \Gamma_\phi)_f \vec{S} \cdot (\nabla \phi)_f. \quad (3.19)$$

Nel caso generico di mesh non ortogonale, si opera una suddivisione del termine  $\vec{S} \cdot (\nabla\phi)_f$  in un contributo ortogonale e in uno non ortogonale

$$\vec{S} \cdot (\nabla\phi)_f = \vec{\Delta} \cdot (\nabla\phi)_f + \vec{k} \cdot (\nabla\phi)_f. \quad (3.20)$$

I termini introdotti nella (3.20) devono soddisfare la relazione

$$\vec{S} = \vec{\Delta} + \vec{k}. \quad (3.21)$$

Ora, si utilizza il metodo "Minimum correction approach", visibile all'interno della Figura 3.3, secondo il quale la decomposizione del vettore superficie deve essere eseguita in modo tale per cui i vettori  $\vec{\Delta}$  e  $\vec{k}$  siano fra di loro ortogonali. Viene dunque calcolato  $\vec{k}$  dall'equazione (3.21) e  $\vec{\Delta}$  dalla relazione

$$\vec{\Delta} = \frac{\vec{d} \cdot \vec{S}}{\vec{d} \cdot \vec{d}} \vec{d} \quad (3.22)$$

Per concludere, la forma finale discretizzata del termine diffusivo si esprime come

$$\vec{S} \cdot (\nabla\phi)_f = |\vec{\Delta}| \frac{\phi_N - \phi_P}{|\vec{d}|} + \vec{k} \cdot (\nabla\phi)_f \quad (3.23)$$

dove

$$(\nabla\phi)_f = f_x(\nabla\phi)_P + (1 - f_x)(\nabla\phi)_N \quad (3.24)$$

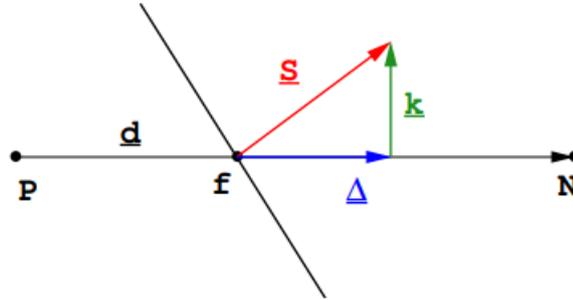


Figura 3.3: Minimum Correction Approach.

### 3.3.3 Discretizzazione del termine di generazione

Per quanto riguarda il termine di generazione, esso viene trattato come funzione di  $\phi$  e linearizzato come

$$S_\phi(\phi) = Su + Sp\phi \quad (3.25)$$

dove  $S_p$  e  $S_u$  dipendono da  $\phi$ . Si può, inoltre, esprimere l'integrale sul volume di controllo come

$$\int_{V_P} S_\phi(\phi) dV = S_u V_P + S_p \phi V_P \quad (3.26)$$

Ora, utilizzando le equazioni (3.5), (3.23) e (3.26) è possibile scrivere l'equazione (3.2) nella forma semi-discretizzata

$$\begin{aligned} \int_t^{t+\Delta t} \left( \frac{\partial \rho \phi}{\partial t} \right)_P V_P + \sum_f F \phi_f - \sum_f (\rho \Gamma_\phi)_f \vec{S} \cdot (\nabla \phi)_f dt \\ = \int_t^{t+\Delta t} (S_u V_P + S_p V_P \phi_P) dt \end{aligned} \quad (3.27)$$

### 3.3.4 Discretizzazione del termine temporale

Lo step successivo riguarda la discretizzazione temporale. In primo luogo si consideri una variazione temporale del tipo

$$\phi(t + \Delta t) = \phi^t + \Delta t \left( \frac{\partial \phi}{\partial t} \right)^t \quad (3.28)$$

dove

$$\phi(t) = \phi^t \quad (3.29)$$

Allora la derivata nel tempo può essere espressa come

$$\left( \frac{\partial \rho \phi}{\partial t} \right)_P = \frac{\rho_P^n \phi_P^n - \rho_P^0 \phi_P^0}{\Delta t} \quad (3.30)$$

dove

$$\phi^n = \phi(t + \Delta t) \quad (3.31)$$

$$\phi^0 = \phi(t) \quad (3.32)$$

La nuova forma dell'equazione di trasporto discretizzata diventa di ordine di accuratezza 1 nel tempo e una scelta deve essere fatta sul come valutare i valori nel centro della faccia di  $\phi$  e  $\nabla \phi$ .

$$\begin{aligned} \frac{\rho_P^n \phi_P^n - \rho_P^0 \phi_P^0}{\Delta t} V_P + \sum_f F \phi_f - \sum_f (\rho \Gamma_\phi)_f \vec{S} \cdot (\nabla \phi)_f \\ = S_u V_P + S_p V_P \phi_p \end{aligned} \quad (3.33)$$

Una scelta possibile è quella di utilizzare un metodo come quello Eulero implicito, il quale esprime il valore al centro della faccia come funzione dei valori aggiornati nel tempo al centro delle celle

$$\phi_f = f_x \phi_P^n + (1 - f_x) \phi_N^n \quad (3.34)$$

$$\vec{S} \cdot (\nabla\phi)_f = |\vec{\Delta}| \frac{\phi_N^n - \phi_P^n}{|\vec{d}|} + \vec{k} \cdot (\nabla\phi)_f \quad (3.35)$$

L'utilizzo di un metodo implicito garantisce la limitatezza della soluzione ed è incondizionatamente stabile.

### 3.3.5 Sistema algebrico risolutivo

L'obiettivo è la determinazione del valore aggiornato nel tempo di  $\phi_P$ . Siccome  $\phi_f$  e  $(\nabla\phi)_f$  dipendono dal valore di  $\phi$  nelle celle confinanti, l'utilizzo del metodo Eulero implicito produce un'equazione algebrica del tipo

$$a_P \phi_P^n + \sum_N a_N \phi_N^n = R_p \quad (3.36)$$

Per ciascun volume di controllo si crea un'equazione di questa forma. Il valore di  $\phi_P^n$  dipende dai valori nelle celle confinanti e quindi si crea un sistema di equazioni algebriche del tipo

$$[A][\phi] = [R] \quad (3.37)$$

dove  $[A]$  è una matrice sparsa con coefficienti  $a_P$  sulla diagonale e  $a_N$  fuori diagonale,  $[\phi]$  è il vettore contenente i  $\phi$  per ogni volume di controllo e  $[R]$  è il vettore dei termini noti. Quando questo sistema viene risolto si ottiene un nuovo set di valori  $\phi$ , ovvero la soluzione del nuovo time-step.

## 3.4 Descrizione di OpenFOAM

OpenFOAM (Field Operation and Manipulation) è una libreria C++ sviluppata da OpenCFD Ltd all'interno di ESI Group e distribuita da OpenFOAM Foundation. Esso implementa algoritmi in grado di risolvere equazioni differenziali alle derivate parziali su domini arbitrariamente complessi e trova impiego soprattutto nel campo della fluidodinamica computazionale (CFD). Esso viene sfruttato per creare eseguibili, conosciuti come **applications**. Gli **applications** si dividono in due categorie:

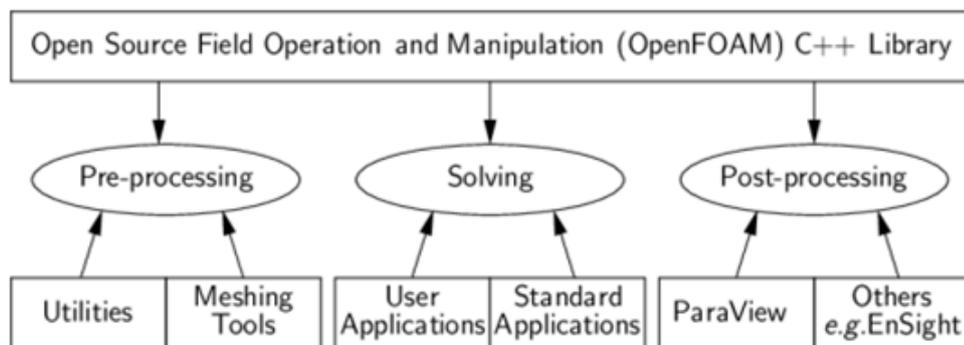
- **solvers**, ovvero algoritmi progettati per risolvere problemi specifici all'interno della meccanica del continuo
- **utilities**, ovvero programmi in grado di adempiere compiti riguardanti la manipolazione dei dati, la creazione di griglie computazionali e il post-processing.

All'interno dei **solvers** è possibile trovare algoritmi risolutivi riguardanti moti comprimibili ed incompressibili, reazioni chimiche e combustioni, elettromagnetismo, DNS (direct numerical simulation), trasferimento di calore, modellazione multifase, stress-analysis e finanza.

Le librerie all'interno della suite sono dinamicamente collegate fra loro durante la compilazione dei **solvers** e delle **utilities**.

OpenFOAM è fornito con ambienti dedicati al pre- e post-processing.

Essendo OpenFOAM un pacchetto software open-source, ovvero con codice sorgente disponibile, l'utente è in grado di creare nuovi **solvers** e nuove **utilities**. Grazie a ciò, esso è ampiamente impiegato in ambito accademico e nell'industria di processo. La struttura complessiva di OpenFOAM è mostrata in Figura 3.4:

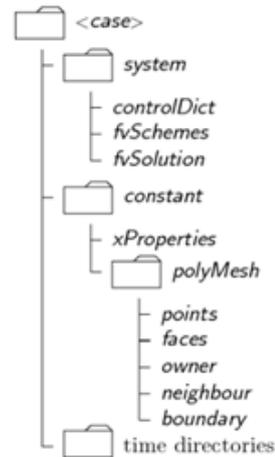


**Figura 3.4:** Struttura della libreria OpenFOAM.

La struttura di una directory base (**case**) per la simulazione di un caso in OpenFOAM, contenente il set minimo di files richiesti per la compilazione di un' **application**, è mostrato in Figura 3.5.

### 3.4.1 Directory constant

Questa directory contiene una descrizione completa del dominio fluido e del dominio fluido nella sub-directory **polymesh**. Inoltre, all'interno di **constant** vengono anche specificate le proprietà fisiche e di trasporto del fluido, ad esempio nel file **transportProperties**. Inoltre, qualora fosse necessario l'impiego di un modello di turbolenza, possono esserci file specificanti i parametri del modello di turbolenza, ad esempio nel file **RASproperties**.



**Figura 3.5:** Struttura di una generica directory base di simulazione in OpenFOAM.

### 3.4.2 Directory system

Essa permette di settare i parametri associati all'algoritmo e alla procedura risolutiva.

La directory `system` deve almeno contenere i seguenti 3 files:

- `controlDict`: vengono settati i parametri per la compilazione del caso includendo il `solver` risolutivo, il tempo di inizio e di fine, il passo temporale e alcuni parametri legati all'output dei risultati
- `fvSchemes`: vengono indicati gli schemi di discretizzazione e gli schemi numerici impiegati al fine di risolvere numericamente le equazioni di trasporto. Viene specificato lo schema numerico adottato per ogni termine delle equazioni da risolvere
- `fvSolution`: vengono settati i parametri di controllo riguardanti i solutori delle equazioni, le tolleranze per arrivare a convergenza e gli algoritmi risolutivi.

### 3.4.3 Directories time

Esse sono cartelle etichettate con un numero indicante il un tempo specifico. Esse vengono stampate in automatico e contengono ad esempio la descrizione dei campi di velocità, pressione e frazione volumica all'interno della simulazione in un istante di tempo ben preciso. A inizio simulazione deve essere presente la directory `0`, la quale specifica le condizioni al contorno e iniziali del calcolo. All'interno di OpenFOAM ogni campo deve essere sempre inizializzato, anche quando la soluzione non lo richiede strettamente.

### 3.5 Descrizione del solver interFoam

InterFoam, come è scritto all'interno della user guide di OpenFOAM, è un solver per due fluidi immiscibili, incomprimibili, isoterma che utilizza il metodo di tracciamento dell'interfaccia volume fractions VOF (Volume of fluid method).

Una volta installato OpenFOAM, all'interno della directory applications è possibile trovare tutti i solvers presenti nella libreria. In seguito viene riportato e commentato il codice C++ riguardante il solver interFoam.

```

1  /*-----*\
2  ===== |
3  \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \\      / O p e r a t i o n      | Website: https://openfoam.org
5  \\      / A n d      | Copyright (C) 2011-2019 OpenFOAM Foundation
6  \\/      M a n i p u l a t i o n      |
7  -----*
8  License
9      This file is part of OpenFOAM.
10     OpenFOAM is free software: you can redistribute it and/or modify it
11     under the terms of the GNU General Public License as published by
12     the Free Software Foundation, either version 3 of the License, or
13     (at your option) any later version.
14     OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
15     ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
16     FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
17     for more details.
18     You should have received a copy of the GNU General Public License
19     along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
20 Application
21     interFoam
22 Description
23     Solver for 2 incompressible, isothermal immiscible fluids using a VOF
24     (volume of fluid) phase-fraction based interface capturing approach,
25     with optional mesh motion and mesh topology changes including adaptive
26     re-meshing.
27  \*-----*/
28
29 #include "fvCFD.H"
30 #include "dynamicFvMesh.H"
31 #include "CMULES.H"
32 #include "EulerDdtScheme.H"
33 #include "localEulerDdtScheme.H"
34 #include "CrankNicolsonDdtScheme.H"
35 #include "subCycle.H"
36 #include "immiscibleIncompressibleTwoPhaseMixture.H"

```



```

82     #include "setDeltaT.H"
83 }
84
85     runTime++;
86
87     Info<< "Time = " << runTime.timeName() << nl << endl;
88
89     // --- Pressure-velocity PIMPLE corrector loop
90     while (pimple.loop())
91     {
92         if (pimple.firstPimpleIter() || moveMeshOuterCorrectors)
93         {
94             mesh.update();
95
96             if (mesh.changing())
97             {
98                 // Do not apply previous time-step mesh compression flux
99                 // if the mesh topology changed
100                if (mesh.topoChanging())
101                {
102                    talphaPhi1Corr0.clear();
103                }
104
105                gh = (g & mesh.C()) - ghRef;
106                ghf = (g & mesh.Cf()) - ghRef;
107
108                MRF.update();
109
110                if (correctPhi)
111                {
112                    // Calculate absolute flux
113                    // from the mapped surface velocity
114                    phi = mesh.Sf() & Uf();
115
116                    #include "correctPhi.H"
117
118                    // Make the flux relative to the mesh motion
119                    fvc::makeRelative(phi, U);
120                }
121
122                mixture.correct();
123
124                if (checkMeshCourantNo)
125                {
126                    #include "meshCourantNo.H"

```

```

127         }
128     }
129 }
130
131 #include "alphaControls.H"
132 #include "alphaEqnSubCycle.H"
133
134 mixture.correct();
135
136 #include "UEqn.H"
137
138 // --- Pressure corrector loop
139 while (pimple.correct())
140 {
141     #include "pEqn.H"
142 }
143
144 if (pimple.turbCorr())
145 {
146     turbulence->correct();
147 }
148 }
149
150 runTime.write();
151
152 Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
153     << "   ClockTime = " << runTime.elapsedClockTime() << " s"
154     << nl << endl;
155 }
156
157 Info<< "End\n" << endl;
158
159 return 0;
160 }
161
162
163 // ***** //

```

- righe 1-27 : questa prima parte è comune a tutti i solver e contiene essenzialmente la descrizione e le finalità del codice
- righe 29-41 : all'interno di questa porzione di codice vengono inclusi nel solver tutti gli "include files". All'interno del file "fvCFD.H" è presente il codice la cui esecuzione applica il metodo ai volumi finiti per discretizzare le equazioni del modello. I files "CrankNicolsonDdtScheme.H" e "EulerDdtScheme.H" indicano

i possibili schemi di discretizzazione temporale applicabili da interFoam. Il file "turbulentTransportModel.H" include una scelta di modelli fra cui quello laminare, che in OpenFOAM è indicato come DNS, RAS (Reynolds Avaraged simulation), LES (Large eddies simulation) e DES (Detached eddy simulation). Il file "fvOptions.H" permette all'utente di inserire termini di generazione all'interno dell'equazione di trasporto della quantità di moto, non utilizzati nel caso della simulazione del riempimento. Il file "pimpleControl.H" richiama l'algoritmo risolutivo PIMPLE, il quale è fusione degli algoritmi PISO e SIMPLE, i quali permettono di risolvere la problematica del pressure-velocity coupling. Riguardo all'algoritmo PIMPLE, esso permette di lavorare anche con un passo temporale alto, non rimanendo quindi all'interno della condizione necessaria per la convergenza numerica espressa dal numero di Courant, in quanto si sfruttano i fattori di rilassamento presenti nell'algoritmo SIMPLE. All'interno delle simulazioni del riempimento, però, la condizione CFL viene sempre rispettata e, dunque, utilizzare l'algoritmo PIMPLE è equivalente ad utilizzare l'algoritmo PISO.

- righe 45-57: viene avviata la funzione principale del solver e tutti i "files.H". Il file "createFields.H" è molto importante in quanto inizializza e crea tutti campi necessari al calcolo, leggendoli dal disco e/o creandoli in memoria.
- righe 61-83: tutti i controlli di tempo vengono attivati, specialmente la funzione "CourantNO.H" che calcola il time step soddisfacendo la relazione CFL

$$Co = \frac{|U|\delta t}{\delta x} < 1. \quad (3.38)$$

In questo modo la stabilità della simulazione viene assicurata.

- riga 85: il tempo viene incrementato e riprodotto a schermo
- righe 92-129: questa parte di codice è destinata ai problemi in cui non si ha una griglia computazionale mobile o adattativa. Nel caso della simulazione del riempimento sono state utilizzate griglie fisse.
- righe 90-131-132: in questa zona le equazioni vengono scritte e risolte. Questa parte viene chiamata "pressure velocity PIMPLE corrector loop". All'inizio di questo ciclo tutte le quantità relative alla frazione di fase sono calcolate e l'equazione di advezione è risolta richiamandola nel file "alphaEqnSubCycle.H" Questo file ha la funzione di processare l'equazione di advezione, all'interno dello stesso intervallo temporale  $\Delta t$ , tante volte quanto specificato dal numero di sottocicli ("nAlphaSubCycles"). Per ciascuno di questi sottocicli il flusso di densità e di velocità vengono inizializzati nuovamente e inseriti di nuovo all'interno del file "alphaEqn.H", all'interno del quale sono presenti le equazioni del modello matematico

- riga 134: tutte le variabili del sistema multifase come  $\rho$ ,  $\nu$ ,  $\mu$  vengono corrette al fine di essere inserite all'interno dell'equazione di trasporto della quantità di moto.
- riga 136: Questa fase del solver PIMPLE, algoritmo semi-implicito per il "pressure velocity-coupling", è chiamata "momentum predictor" ed è introdotta da file "UEqn.H", dove l'equazione della quantità di moto viene implementata e risolta. Questo nome è dovuto al fatto che il campo di velocità ottenuto risolvendo questa equazione non è quello esatto, ma risulta essere solo una approssimazione. Infatti, per risolvere l'equazione viene usata la pressione individuata al time step precedente e quindi il campo di velocità ottenuto non soddisfa l'equazione di continuità, ma soddisfa solo l'equazione di trasporto della quantità di moto. Sapendo che la pressione non può essere calcolata contemporaneamente alla velocità a causa del loro accoppiamento all'interno dell'equazione di trasporto della quantità di moto, è necessario dividere il loro calcolo in tre parti:
  - Viene risolta l'equazione della quantità di moto considerando la pressione calcolata al time step precedente (momentum predictor)
  - Si usa il campo di velocità approssimato calcolato nello step precedente per calcolare la pressione. Viene calcolata la pressione necessaria a rendere il campo di velocità precedentemente calcolato un campo solenoidale, ovvero a divergenza nulla. Questo step viene chiamato pressure solution.
  - Viene aggiunto il gradiente di pressione, calcolata allo step precedente, al campo di velocità approssimato, calcolato allo step 1, al fine di trovare il corretto campo di velocità. Questo step viene chiamato velocity correction.

Quando il terzo passo è raggiunto, il ciclo ricomincia al time step successivo.

- righe 138-148: Il secondo step dell'algoritmo PIMPLE (pressure solution) viene avviato e il campo di pressione viene calcolato risolvendo l'equazione di pressione, ovvero l'equazione di Poisson, come implementato nel file "pEqn.H". Il pressure corrector loop viene svolto tante volte quando specificato nel file "fvSolution" alla voce "nCorrectors", che dice quante volte l'equazione di pressione e la correzione sulla velocità vengono implementate. Dopo che la pressione viene valutata risolvendo un'equazione di Laplace, il campo di velocità è aggiornato e corretto.
- righe 150-155: Quando il loop PIMPLE è terminato, vengono stampati a schermo il tempo di esecuzione e il clock time e si passa all'istante temporale successivo.

Riassumendo brevemente la procedura attuata dal solver interFOAM è possibile sintetizzare la procedura risolutiva nei seguenti step:

- tutte le variabili vengono inizializzate all'interno del dominio fluido e il time step è aggiustato rispettando il vincolo del numero di Courant;
- l'equazione di trasporto per la funzione indicatrice viene risolta. Inoltre vengono aggiornate la densità e la viscosità mediate secondo la funzione indicatrice;
- inizia il ciclo PIMPLE e l'equazione di trasporto della quantità di moto è risolta considerando la pressione del precedente time step;
- la pressione è trovata mediante il loop correttivo di pressione e il campo di velocità viene aggiornato.

All'interno del file "fvOption" sono state inserite le seguenti righe di codice:

```

1 PIMPLE
2 {
3     momentumPredictor no;
4     nNonOrthogonalCorrectors 0;
5     nCorrectors          3;
6     nOuterCorrectors    1;
7 }
```

Disattivando il momentum predictor, la velocità non viene calcolata risolvendo l'equazione di trasporto della quantità di moto con la pressione del precedente timestep, ma si considera direttamente la velocità del precedente timestep nell'equazione di Poisson.

Come già espresso precedentemente, `nCorrectors` rappresenta il numero di cicli del pressure correction loop, in cui in ogni ciclo viene risolta l'equazione di pressione, trovando dunque un nuovo valore di pressione, e viene corretto il campo di velocità. In pratica viene risolta un'equazione del tipo

$$\nabla^2 p = f(U, \nabla p) \tag{3.39}$$

chiamata equazione di Poisson.

Per concludere, `nOuterCorrectors` rappresenta quante volte viene ripetuto un ciclo PIMPLE all'interno dello stesso time step.

# Capitolo 4

## Dettagli numerici e descrizione del caso studiato

### 4.1 Descrizione generale e piano di simulazioni

La simulazione del riempimento viene eseguita adottando come geometria il complesso valvola-bottiglia. Nel processo di riempimento il liquido viene introdotto all'ingresso della valvola. La valvola è di tipo rotativa e permette al film liquido introdotto di acquisire un moto caratterizzato da una componente predominante di swirl. Grazie a ciò, il liquido in bottiglia rimane perfettamente aderente alla parete consentendo un minor intrappolamento (hold-up) di gas all'interno della fase liquida. Ciò accade dal punto di vista sperimentale e l'obiettivo del lavoro di tesi è stato simulare ciò dal punto di vista modellistico. Conoscere il contenuto di gas al termine della fase di riempimento è molto importante in quanto la fase di degasamento immediatamente successiva al riempimento è direttamente connessa all'hold-up di gas all'interno della fase liquida stagnante nel fondo bottiglia. Infatti, durante la prima fase di degasamento si ha formazione di schiuma, la quale altezza massima (picco) aumenta all'aumentare della quantità di gas intrappolato durante la fase di riempimento.

La simulazione del riempimento è stata divisa in due parti. La prima parte riguarderà la simulazione della valvola, mentre la seconda parte riguarderà la simulazione della bottiglia. Simulando la valvola si andranno, dunque, ad estrapolare dai dati in output le condizioni di inlet della simulazione della bottiglia. In pratica, dalle simulazioni della valvola vengono ricavati l'angolo di swirl e la velocità tangenziale relativi al film liquido. Essi serviranno a calcolare le componenti di swirl del film liquido entrante in bottiglia. La procedura di ottenimento dei dati di inlet della bottiglia avviene mediante l'utilizzo del software Paraview, come verrà illustrato in seguito.

Il piano di simulazioni è stato stipulato al fine di simulare il riempimento in presenza di tre diverse portate  $Q_1$ ,  $Q_2$ ,  $Q_3$ . Per ogni portata viene effettuata la simulazione della valvola per un tempo tale per cui il moto sia completamente sviluppato e si raggiunga lo stazionario. La valvola viene poi tagliata ad una certa altezza, ortogonalmente all'asse verticale, e, in questa sezione, vengono calcolate le condizioni di inlet della simulazione della bottiglia. Infatti, terminata la simulazione della valvola, viene impostata la simulazione della bottiglia, la quale zona di inlet è in corrispondenza della precedente sezione di taglio della valvola. Per ogni portata, la simulazione della bottiglia è stata divisa in due fasi:

- fase di riempimento attivo: in questa fase la portata di liquido in inlet è stata posta costante e pari al caso di portata che si vuole simulare. Il tempo di simulazione di questa fase è legato al tempo ricavato dalla prove sperimentali di riempimento;
- fase di riposo: in questa fase la portata di liquido in inlet è stata posta costante e nulla. Il tempo di simulazione di questa fase è legato al tempo ricavato dalla prove sperimentali di riempimento.

I tempi di simulazione per le varie portate sono riportati in tabella 4.1. Terminata la fase di simulazione, si entra in fase di post-processing. Per quanto riguarda la griglia computazionale utilizzata, essa ha subito un vero e proprio percorso di definizione al fine di rendere la simulazione più fisica possibile. In questo capitolo verranno presentate le griglie finali, mentre l'intero percorso che ha portato ad adottare queste meshes verrà ampiamente discusso nel capitolo successivo.

| Portata | Tempo di riempimento attivo | Tempo di riposo |
|---------|-----------------------------|-----------------|
| $Q_1$   | 6,5 sec                     | 0,3 sec         |
| $Q_2$   | 5,9 sec                     | 0,3 sec         |
| $Q_3$   | 5,2 sec                     | 0,3 sec         |

**Tabella 4.1:** Tempo di riempimento attivo e riposo per le tre portate simulate.

#### 4.1.1 Regime fluido adottato all'interno delle simulazioni di valvola e bottiglia

Come spiegato nel capitolo 5, all'interno delle simulazioni si è manifestato un distaccamento del film liquido alla parete della bottiglia. La causa di ciò è stata individuata nel regime fluido utilizzato. Infatti, dal calcolo del numero di Reynolds del film liquido è emerso come per le tre portate simulate si ricada all'interno del

regime turbolento, e ciò è illustrato all'interno del capitolo 5. Dunque, le simulazioni finali della bottiglia sono state realizzate in regime turbolento, adottando il modello di turbolenza  $k-\epsilon$  Standard, proprio per evitare il distacco non fisico del film liquido dalla parete. Per quanto riguarda la valvola, invece, le simulazioni sono state realizzate senza l'utilizzo di un modello di turbolenza e non è stato preso in considerazione un modello di turbolenza. Infatti, benché nelle simulazioni finali della valvola si sia manifestato un distacco del film liquido alla parete della bottiglia, la zona all'interno delle simulazioni della valvola in cui si sono ricavate le condizioni di inlet della simulazioni della bottiglia presenta un film liquido ben aderente alla parete e quindi un comportamento del tutto in linea con ciò che si è osservato sperimentalmente.

## **4.2 Geometria e griglia finali del complesso valvola+bottiglia**

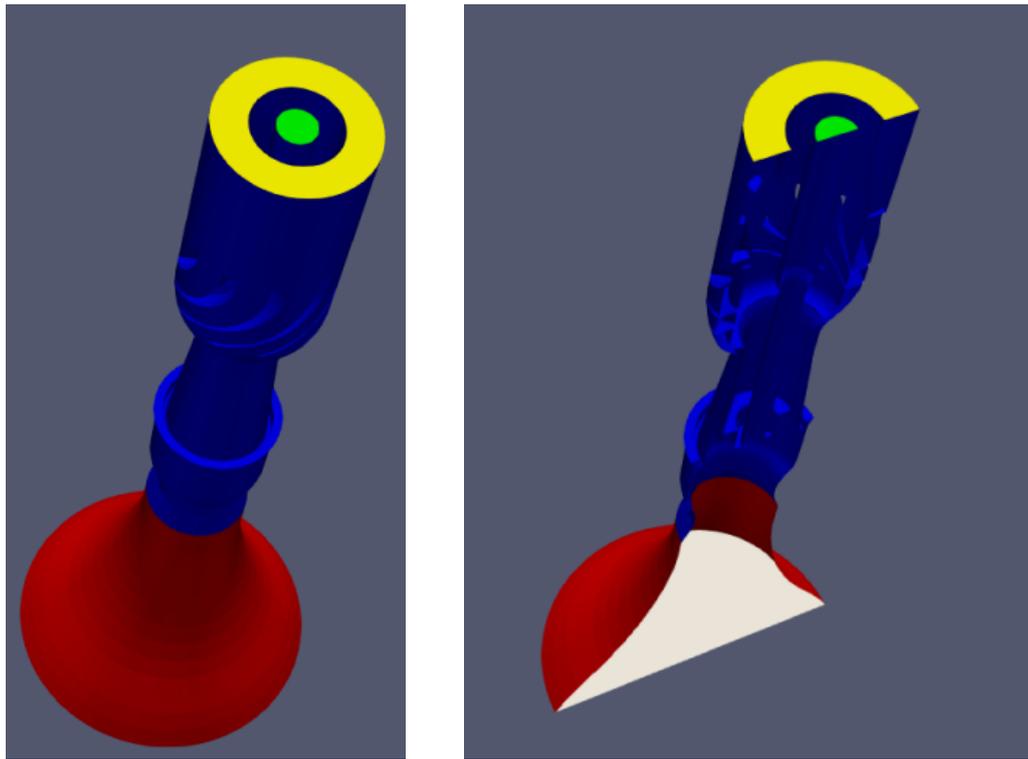
In questa parte della tesi viene riportata la descrizione della geometria finale e della griglia adottata.

### **4.2.1 Geometria e griglia finale: valvola**

In Figura 4.1 è presentata la geometria della simulazione della valvola, la quale comprende anche una porzione di bottiglia. All'interno della descrizione della figura vengono presentate le facce in cui è stata divisa la geometria affinché si possano definire le condizioni al contorno della simulazione. In Figura 4.2 viene illustrata la griglia computazionale adottata per le simulazioni. Questa griglia è composta da 7 milioni di celle e presenta un infittimento uniforme nella zona interna alla valvola e al collo della bottiglia. Ciò permette un'accurata descrizione del film liquido all'ingresso della bottiglia.

### **4.2.2 Geometria e griglia finale: bottiglia**

In Figura 4.3 è presentata la geometria della simulazione della bottiglia. Nella didascalia della figura vengono presentate le facce in cui è stata divisa la geometria affinché si possano definire le condizioni al contorno della simulazione. Per quanto riguarda la griglia computazionale adottata, al fine di descrivere accuratamente il film liquido è stato necessario infittire alle pareti mediante starti di celle ciascuno spesso 0,0002 m. In particolare, come è visibile in Figura 4.4, si sono adottati 10 strati di celle alle pareti. In aggiunta, è presente un infittimento nella zona del tappo

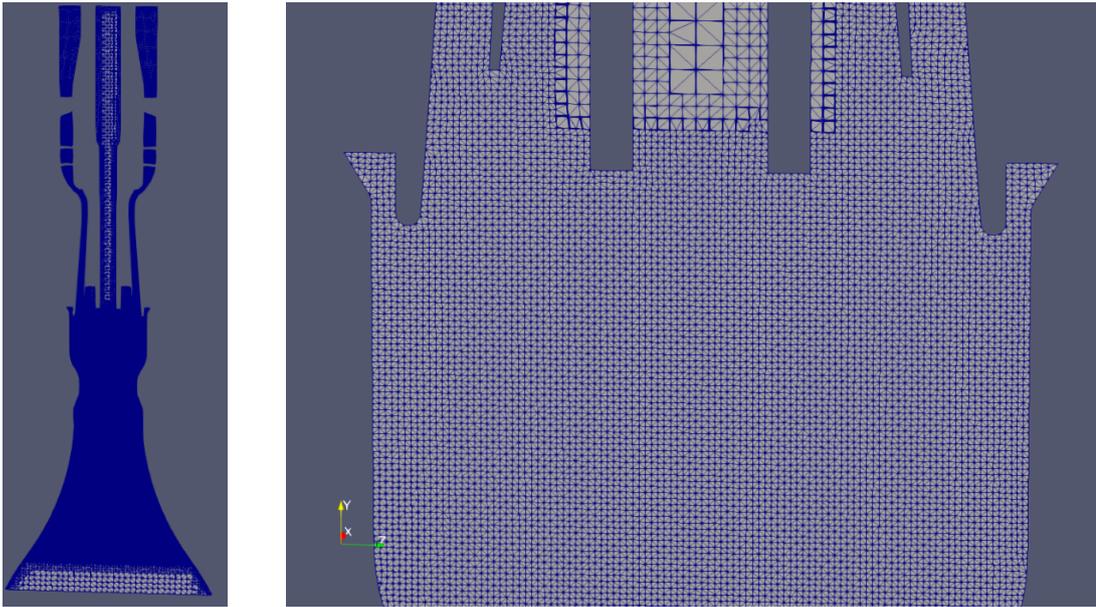


**Figura 4.1:** Sinistra: viene rappresentata la geometria di simulazione della valvola al completo. In verde è evidenziata la sezione di `outlet`, dalla quale uscirà il gas in pressione contenuto all'interno della bottiglia a causa dell'entrata del liquido. In giallo vi è la sezione circolare di ingresso del film liquido: l'`inlet`. In blu sono indicate le pareti della valvola, denominate `Valve_walls`, mentre in rosso è rappresentata una parte della parete della bottiglia, denominata `Top_bottle`. Infine, in bianco è indicata la sezione inferiore della prima parte della bottiglia, chiamata all'interno della simulazione `bottomOutflow`.  
Destra: viene riportata una vista in sezione della valvola

della bottiglia che permette di descrivere l'interfaccia gas-liquido accuratamente. Questa griglia è formata da 6,3 milioni di celle.

### 4.3 Condizioni al contorno e iniziali riguardanti la simulazione della valvola: cartella 0

Come già espresso, la cartella 0 contiene le condizioni iniziali e al contorno per tutte le variabili primarie. Nel caso della valvola vengono date condizioni al contorno e iniziali riguardanti la velocità (file `U`), la frazione volumica (`alpha.water.origin`),

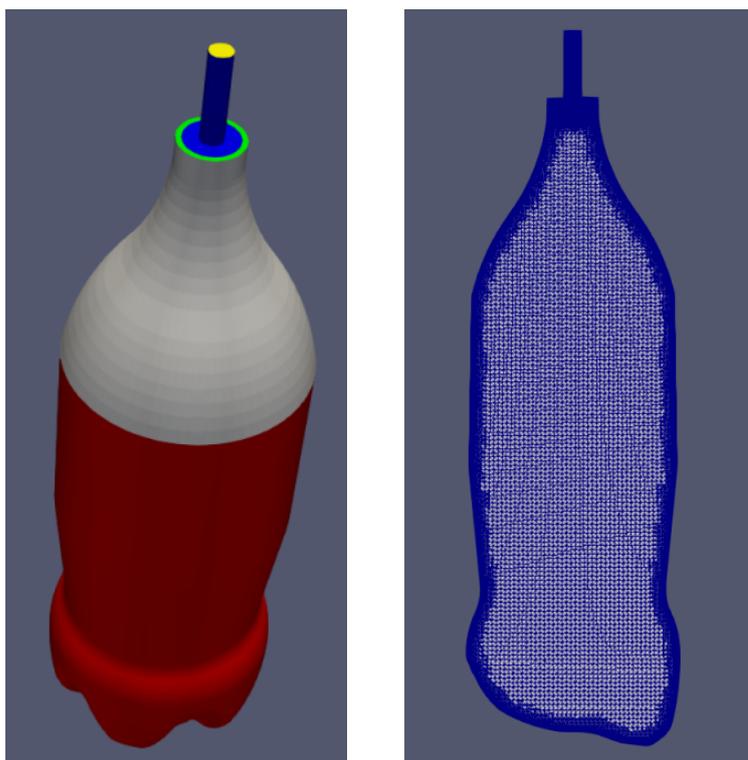


**Figura 4.2:** Sinistra: sezione della zona centrale della mesh. Destra: particolare della sezione nella zona terminale della valvola. Si noti l'infittimento uniforme costituito da celle molto piccole

la pressione ( $p_{rgh}$ ).

### 4.3.1 File U

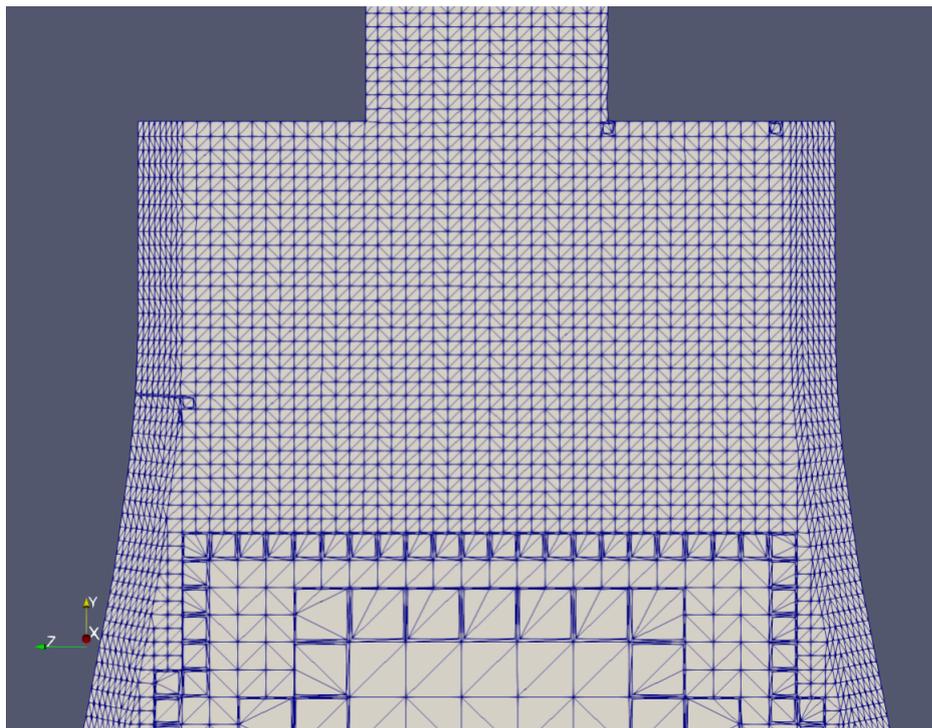
All'interno di esso viene stabilito che la variabile  $U$  ha la dimensione di una velocità. Inoltre viene inizializzato il dominio computazionale ad un certo valore vettoriale con il comando `internalField`. Viene, poi, fissato il valore di velocità costante (`fixedValue`) nella sezione di inlet. Per quanto riguarda le patches `outlet` e `bottomOutflow`, esse devono essere libere, ovvero permettere sia l'inflow che l'outflow in accordo con il flusso interno. Per esprimere ciò viene usato il comando `pressureInletOutletVelocity`, il quale applica gradiente nullo a tutte le componenti, eccetto dove è presente inflow. In questo caso viene applicata una condizione `fixedvalue` alla componente tangenziale. Per quanto riguarda le pareti, viene assegnata una condizione di `noSlip`, la quale setta velocità nulla in corrispondenza della patch `wall`.



**Figura 4.3:** Sinistra: viene rappresentata la geometria di simulazione della bottiglia al completo. In giallo è presente la sezione di `outlet`, dalla quale uscirà il gas in pressione contenuto all'interno della bottiglia a causa dell'entrata del liquido. In verde vi è la sezione di circolare di ingresso del film liquido: l'`inlet`. Questa corona circolare è utilizzata al fine di avere già all'ingresso della bottiglia il film liquido attaccato alla parete, come accade nelle prove sperimentali. In blu viene indicata la parete `neck`, la quale è un artificio di simulazione la cui utilità viene descritta all'interno del capitolo 5. Infine, le parete della bottiglia è stata divisa in due sezione: in bianco `Top_bottle` e in rosso `Bottom_bottle`. Questa suddivisione della parete è finalizzata alla realizzazione degli strati di celle denominati `layers` che permettono di descrivere accuratamente il film liquido alla parete. Destra: sezione della griglia computazionale adottata. Si noti l'infittimento alle pareti e nella zona del tappo. Qui, infatti, è necessario descrivere accuratamente l'interfaccia gas-liquido.

### 4.3.2 File `p_rgh`

All'interno di esso viene stabilito che la variabile `p_rgh` ha la dimensione di una pressione. In particolare, `p_rgh` è la pressione epurata dal contributo idrostatico. Dove `p_rgh` non è fissata si impone la boundary condition `fixedFluxPressure`,



**Figura 4.4:** Particolare della zona del tappo all'interno della griglia computazionale della bottiglia. Si notino i 10 **layers** attaccati alla parete e la **box** d'infiltramento.

la quale setta il gradiente di pressione ad un valore tale che il flusso sia quello specificato dalla condizione al contorno sulla velocità. Per quanto riguarda le patches **bottomOutflow** e **outlet**, la condizione definita è quella di pressione totale. Questa condizione prevede che la pressione statica alla patch  $p_p$  rispetti la condizione

$$p_p = p_0 - \frac{1}{2}|U|^2 \quad (4.1)$$

dove  $p_0$  è la pressione totale. Questa condizione, dunque, richiede in input il valore di  $p_0$ , che è pari alla pressione a cui avviene il riempimento.

### 4.3.3 File `alpha.water.orig`

È necessario specificare una condizione iniziale non uniforme per la frazione di fase. Questo può essere fatto mediante il comando `setFields`. Esso richiede un dizionario `setFieldsDict`, localizzato nella directory `system`. In questo modo è possibile definire le regioni del dominio fluido in cui all'inizio è presente il fluido. Nel caso specifico della simulazione della valvola, la regione iniziale dove è presente il liquido ( $\alpha = 1$ ) è un'anello cilindrico. Per quanto riguarda le

condizioni al contorno, all'inlet con la dicitura `alpha_in` si richiama il file `boundary`, situato nella directory `include`, all'interno del quale viene specificato `alpha_in=1`. Inoltre, con la dicitura `inletOutlet` si fissa una condizione `zeroGradient`, ovvero di gradiente nullo della variabile, quando il flusso è rivolto verso l'esterno, mentre si applica una condizione `fixedValue`, ovvero di valore costante della variabile, quando il flusso è rivolto verso l'interno. Per quanto riguarda le pareti, viene applicato il comando `constantAlphaContactAngle`, il quale richiede che l'utente specifichi un angolo di contatto statico, `theta0`.

All'interno delle simulazioni, è stata valutata anche l'opzione `zeroGradient` per quanto riguarda le pareti, ma essa è stata scartata in quanto inserendo gli angoli di contatto si è ottenuta una descrizione d'interfaccia più dettagliata.

## 4.4 Condizioni al contorno e iniziali riguardanti la simulazione della bottiglia: cartella 0

Come già espresso, la cartella 0 contiene le condizioni iniziali e al contorno per tutte le variabili primarie e per ogni patch definita. Nel caso della bottiglia vengono date condizioni al contorno e iniziali riguardanti la velocità (file `U`), la frazione volumica (`alpha.water.origin`), la pressione (`p_rgh`), la viscosità turbolenta (`nut`), la velocità di dissipazione dell'energia cinetica turbolenta (`epsilon`) e per l'energia cinetica turbolenta (`k`).

### 4.4.1 File `U`

All'interno della simulazione della bottiglia, le condizioni nella sezione di `outlet` e alle pareti sono fornite in modo del tutto analogo alla valvola. Per quanto riguarda la sezione di `inlet`, la condizione è data tramite il comando `swirlFlowRateInletVelocity`. Esso permette di inserire un flusso in ingresso portata costante e moto rotatorio. Dunque, è necessario specificare anche l'asse attorno al quale si sviluppa il moto rotatorio del liquido in ingresso.

### 4.4.2 File `p_rgh`

Per quanto riguarda le specifiche sulla pressione, viene utilizzato il comando `fixedFluxPressure` per le patches `inlet`, `bottom_bottle`, `top_bottle`, `neck`. Per quanto riguarda l'`outlet` si usa un comando `totalPressure`.

### 4.4.3 File `alpha.water.orig`

La frazione di fase subisce lo stesso trattamento ricevuto nella simulazione della valvola. Infatti, anche qui l'utilizzo degli angoli di contatto permette di avere maggiore risoluzione del film liquido. In `setFieldsDict`, il campo di  $\alpha$  viene inizializzato ponendo anche qui il liquido in un anello cilindrico nella zona di `inlet`.

### 4.4.4 File `nut`

Una serie di diversi modelli di wall functions sono applicabili in OpenFOAM. Esse agiscono da condizioni al contorno. OpenFOAM permette di applicare diverse wall functions alle diverse patches. La scelta del modello di wall functions viene specificata all'interno del file `nut`, ovvero il campo di viscosità turbolenta  $\nu_t$ . All'interno di questo file viene associato alle pareti il comando `nutkWallFunction`, che fornisce un vincolo di parete per la viscosità turbolenta. Le equazioni del modello non vengono riportate in questa sede, ma sono reperibili nella sezione "epsilonWallFunction" all'interno della guida di OpenFOAM.

### 4.4.5 File `epsilon`

All'interno di questo file viene associato alle pareti il comando `epsilonWallFunction`, che fornisce un vincolo di parete per la velocità di dissipazione dell'energia cinetica turbolenta. Le equazioni del modello non vengono riportate in questa sede, ma sono reperibili nella sezione "epsilonWallFunction" all'interno della guida di OpenFOAM.

### 4.4.6 File `k`

All'interno di questo file viene associato alle pareti il comando `kqRWallFunction`, che fornisce le condizioni al contorno per l'energia cinetica turbolenta tramite una sorta di involucro attorno alla condizione `zeroGradient`. Inoltre, questo comando implementa anche le condizioni al contorno la radice delle energia cinetica turbolenta e per il tensore degli sforzi di Reynolds. Le equazioni del modello non vengono riportate in questa sede, ma sono reperibili nella sezione "kqRWallFunction" all'interno della guida di OpenFOAM.

## 4.5 Dettagli riguardanti la simulazione della bottiglia: wall functions

Per quanto riguarda la simulazione della bottiglia, come già accennato, alle pareti vengono applicate le wall functions. In questa sezione esse vengono contestualizzate ed introdotte.

Le wall functions vengono usate per descrivere accuratamente il profilo di flusso in prossimità della parete in quanto il modello di turbolenza si applica solo a campi completamente sviluppati. Esse forniscono le condizioni al contorno nella regione prossima alla parete per le equazioni di trasporto della quantità di moto e della turbolenza. Quando vengono utilizzate le wall functions non è necessario risolvere il boundary layer, che richiederebbe un notevole aumento delle celle della griglia computazionale in prossimità della parete. Infatti, i gradienti di velocità vicino alla pareti sono grandi e bisognerebbe usare molte celle, ovvero un'alta risoluzione di griglia, per risolvere questi gradienti in modo accurato. L'intento delle wall functions è rimpiazzare questo grande numero di celle con una cella sola. In questo modo aumenta la stabilità del codice e inoltre si riducono i tempi di simulazione. In aggiunta, è possibile dire che le wall functions sono funzioni empiriche che riproducono il comportamento osservato vicino alla parete.

Per sapere in quali condizioni è possibile applicare le wall functions, si introducono i seguenti parametri

$$y^+ = \frac{yU_t}{\nu} \quad (4.2)$$

$$U_t = \sqrt{\frac{\tau_w}{\rho}} \quad (4.3)$$

dove  $U_t$  è detta "friction velocity",  $\tau_w$  è lo sforzo di taglio alla parete,  $y$  è la distanza dalla parete e  $y^+$  è una distanza adimensionale. Il valore di  $y^+$  relativo alla prima cella è fondamentale in quanto è indice della locazione della prima cella. Inoltre è possibile definire la velocità adimensionale come

$$U^+ = \frac{U}{U_t} \quad (4.4)$$

La regione vicino alla parete si divide in tre zone: il sottostrato viscoso, lo strato buffer e la regione logaritmica.

- Sottostrato viscoso (o viscous sublayer) ( $y^+ < 5$ )  
All'interno del sottostrato viscoso domina l'effetto viscoso e quindi si può assumere che lo sforzo di taglio del fluido sia uguale allo sforzo di taglio alla parete  $\tau_w$ . All'interno di questa regione si ha un profilo di velocità lineare dato da

$$U^+ = y^+ \quad (4.5)$$

La linea blu nel viscous sub-layer in Figura 4.5 rappresenta la relazione lineare.

- Strato buffer (o buffer layer) ( $5 < y^+ < 30$ )  
In questo strato gli sforzi viscosi e turbolenti sono di ordine di grandezza simile e a causa di ciò il profilo di velocità non è ben definito. In OpenFOAM il

buffer layer può essere trattato con due diversi approcci. Il primo applica la legge lineare del sub-strato viscoso anche in questa regione, il secondo applica la legge logaritmica dello strato logaritmico.

- Strato logaritmico (o logarithmic area) ( $30 < y^+ < 200$ )  
In questa regione le forze inerziali sono dominanti. Si ha un profilo di velocità del tipo

$$U^+ = \frac{1}{\kappa} \ln Ey^+ \quad (4.6)$$

dove  $\kappa$  è la costante di Von Karman uguale a 0,41 e  $E$  rappresenta un parametro di parete che viene posto pari a 9,81 per pareti lisce. Nella Figura 4.5 la linea rossa nella regione logaritmica rappresenta l'andamento all'interno della suddetta regione.

All'interno del codice CFD OpenFOAM, il valore di  $y^+$  viene calcolato riferendosi alla distanza fra il centro della prima cella alla parete e la parete stessa. Dalla definizione di  $y^+$  emerge come esso vari nel tempo con la simulazione: il codice CFD valuta la  $y^+$  continuamente e poi sceglie la funzione adatta a seconda del valore calcolato. L'implementazione delle wall functions è, dunque, definita automatica. Ora, si ha che la velocità alla parete è nulla (data dalla condizione `noslip`) e la velocità al centro della prima cella  $U_p$  è calcolata dall'equazione di trasporto della quantità di moto o dalla precedente iterazione o dalla condizione iniziale. Per fare ciò diventa necessario calcolare lo sforzo di taglio alla parete. Se la variazione attraverso la cella è lineare (viscous sub layer) allora lo sforzo di taglio alla parete è calcolabile come

$$\tau_w = \nu \left( \frac{\partial U}{\partial y} \right)_{y=0} = \nu \frac{U_p - 0}{\Delta y_p} = \nu \frac{U_p}{\Delta y_p} \quad (4.7)$$

dove  $\Delta y_p$  è la distanza dalla parete del punto P al centro della prima cella. Se la variazione attraverso la cella è non lineare (log law region) allora lo sforzo di taglio alla parete è calcolabile come

$$\tau_w = \nu \left( \frac{\partial U}{\partial y} \right)_{y=0} = \frac{U_t U_p}{\frac{1}{\kappa} \ln(Ey^+)} \quad (4.8)$$

All'interno del codice CFD viene utilizzato uno schema del secondo ordine in cui la variazione attraverso la cella è lineare. Allora, se si è nella regione logaritmica, lo sforzo di taglio alla parete viene calcolato in modo errato. A questo proposito viene introdotta la viscosità a parete  $\nu_w$ , definita come

$$\nu_w = \nu + \nu_t \quad (4.9)$$

Proseguendo, dall'uguaglianza

$$\nu_w \frac{U_p}{y_p} = \frac{U_t U_p}{\frac{1}{\kappa} \ln(Ey^+)} \quad (4.10)$$

si ha che

$$\nu_w = \frac{U_t \Delta y_p}{\frac{1}{\kappa} \ln(Ey^+)} \quad (4.11)$$

Il codice modifica solo la viscosità a parete quando viene usata la legge logaritmica delle wall functions. Dunque, viene usata l'espressione

$$\tau_w = \nu_w \frac{U_p}{\Delta y_p} \quad (4.12)$$

dove all'interno del codice si ha che

$$\nu_t = \begin{cases} 0 & \text{se } y^+ < 11,25 \\ \nu \left( \frac{y^+}{\frac{1}{\kappa} \ln(Ey^+)} - 1 \right) & \text{se } y^+ > 11,25. \end{cases} \quad (4.13)$$

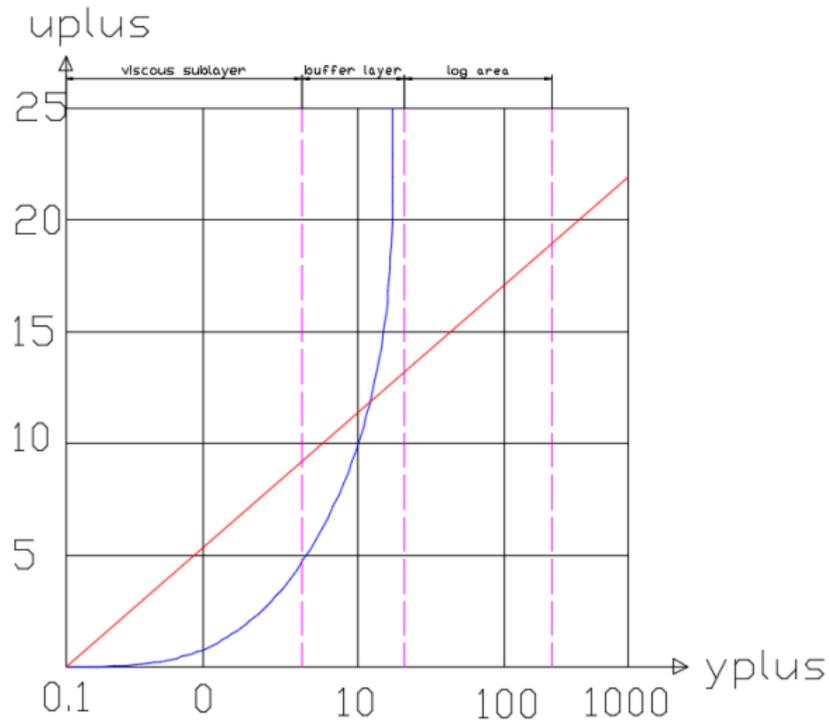
Da ciò emerge come il codice possa scegliere come calcolare lo sforzo di taglio a parete a seconda delle circostanze. All'interno delle simulazioni, quando vengono impiegate le wall functions, è opportuno verificare che valore di  $y^+$  per ogni istante temporale si trovi all'interno del range di validità. Per fare ciò, terminata la simulazione, è opportuno inserire da riga di comando il seguente codice

```
interFoam -postProcess -time -func yPlus
```

## 4.6 Schemi numerici adottati

Il solver `interFOAM` utilizza il metodo MULES (Multidimensional Universal Limiter for Explicit Solution) al fine di mantenere la frazione di fase limitata indipendentemente dagli schemi numerici adottati e dalla struttura della mesh. Grazie a ciò, la scelta riguardante gli schemi di convezione non è limitata solo a schemi fortemente stabili o limitati, ad esempio `upwind`, ma è possibile scegliere fra molteplici alternative. In `OpenFOAM` gli schemi numerici adottati per discretizzare i vari termini delle equazioni differenziali alle derivate parziali vengono descritti nel file `fvSchemes`, il quale è contenuto nella directory `system`. All'interno delle simulazioni finali della valvola, nel file `fvSchemes` sono state inserite le seguenti righe di codice.

```
1 18 ddtSchemes
2 19 {
```



**Figura 4.5:** Relazione fra  $U^+$  e  $y^+$  all'interno delle tre regioni all'interno dell'inner region attaccata alla parete.

```

3 20     default      Euler;
4 21 }
5 22
6 23 gradSchemes
7 24 {
8 25     default      Gauss linear;
9 26 }
10 27
11 28 divSchemes
12 29 {
13 30     div(rhoPhi,U) Gauss linearUpwind grad(U);
14 31     div(phi,alpha) Gauss vanLeer;
15 32     div(phirb,alpha) Gauss linear;
16 33     div(((rho*nuEff)*dev2(T(grad(U)))) Gauss linear;
17 34 }
18 35
19 36 laplacianSchemes
20 37 {

```

```

21 38     default      Gauss linear corrected;
22 39  }
23 40
24 41  interpolationSchemes
25 42  {
26 43     default      linear;
27 44  }
28 45
29 46  snGradSchemes
30 47  {
31 48     default      corrected;
32 49  }
33 50
34 51
35 52  // ***** //

```

Per quanto riguarda la simulazione finale della bottiglia, all'interno della sezione `divSchemes` sono state aggiunte le righe

```

1  div(phi,k) Gauss upwind;
2  div(phi,epsilon) Gauss upwind;

```

al fine del tenere conto del modello di turbolenza utilizzato.

All'interno del file, è possibile osservare come venga adottato una schema di tipo Eulero implicito del primo ordine al fine di discretizzare i termini temporali (righe 18-21).

Per quanto riguarda i termini spaziali, come già descritto nel capitolo 3, viene utilizzato il teorema di Gauss al fine di trasformare l'integrale sul volume di controllo in una somma lungo le facce del volume di controllo. La somma sulle facce del volume di controllo richiede la conoscenza del valore della proprietà che si sta discretizzando al centro di ogni faccia. Se si indica con  $\Phi$  una generica proprietà e con  $\Phi_f$  il valore di quella proprietà al centro della faccia allora, al fine di ottenere la discretizzazione mediante il solo uso dei valori al centro delle celle confinanti con la faccia  $\Phi_P$  e  $\Phi_N$ , è necessario specificare la relazione

$$\Phi_f = f(\Phi_P, \Phi_N) \quad (4.14)$$

La funzione  $f$  è una funzione di interpolazione. All'interno di OpenFOAM sono disponibili diverse funzioni di interpolazione.

Per quanto riguarda la discretizzazione dei termini di gradiente ( $\nabla$ ), il valore della proprietà, contenuta all'interno dell'operatore gradiente, al centro della faccia è ottenuta per interpolazione lineare (central differencing), espresso dal comando `linear`. Esso è uno schema di secondo ordine.

All'interno della sezione `divSchemes` viene presentata la discretizzazione dei termini di divergenza ( $\nabla \cdot$ ).

- `div(rhoPhi,U)`: rappresenta il termine  $\nabla \cdot (\rho \vec{U} \vec{U})$ . L'interpolazione è del tipo `linearUpwind grad(U)`, il quale è uno schema del secondo ordine non limitato. Questo schema applica un'interpolazione pesata, con una correzione esplicita basata sul gradiente locale di velocità di cella.
- `div(phi,alpha)`: rappresenta il termine  $\nabla \cdot (\vec{U} \alpha)$ . L'interpolazione è del tipo `vanLeer`, il quale è uno schema di secondo ordine limitato.
- `div(phiRb,alpha)`: rappresenta il termine  $\nabla \cdot (\vec{U}_{rb} \alpha)$ . L'interpolazione è del tipo `linear`, il quale è uno schema di secondo ordine non limitato adatto per griglie isotropiche.
- `div(((rho*nuEff)*dev2(T(grad(U)))))`: rappresenta il termine  $\nabla \cdot (\rho(\nu + \nu_t)((\nabla \vec{U})^T - \frac{2}{3}(\nabla \cdot \vec{U})\mathbf{I}))$ . L'interpolazione è del tipo `linear`.
- `div(phi,k)`: rappresenta il termine  $\nabla \cdot (\vec{U} k)$ . L'interpolazione è del tipo `upwind`, il quale è uno schema di primo ordine limitato, in cui il valore alla faccia è dato dal valore al centro della cella upstream.
- `div(phi,epsilon)`: rappresenta il termine  $\nabla \cdot (\vec{U} \epsilon)$ . L'interpolazione è del tipo `upwind`.

Analizzando, ora, la sezione `laplacianSchemes` si prendono in esame i termini contenuti nell'operatore laplaciano ( $\Delta$ ). Questi termini vengono discretizzati con uno schema del secondo ordine non limitato conservativo.

Il comando `interpolationSchemes` indica che tutti i termini non specificati vengano interpolati con uno schema `linear`. Per approfondimenti consultare la guida di OpenFOAM alla sezione "Numerical schemes".

## 4.7 Proprietà dei fluidi

Esse vengono indicate all'interno del dizionario `transportProperties` contenuto nella directory `constant`. In questo file è possibile nominare i due fluidi e indicare per ciascun fluido la reologia e le proprietà caratterizzanti come densità, viscosità cinematica e tensione superficiale fra i due fluidi. All'interno della Tabella 4.2 vengono riportate le proprietà fluide adottate all'interno della simulazione finale di valvola e bottiglia. Esse sono riferite alla pressione del riempimento  $P_{tank}$  e alla temperatura di 20°C.

|   |                        |                           |
|---|------------------------|---------------------------|
| Reologia acqua                              | Fluido Newtoniano      |                           |
| Viscosità cinematica acqua                  | $1,053 \times 10^{-6}$ | $\text{m}^2\text{s}^{-1}$ |
| Densità acqua                               | 998,826                | $\text{kg m}^{-3}$        |
| Reologia CO <sub>2</sub>                    | Fluido Newtoniano      |                           |
| Viscosità cinematica CO <sub>2</sub>        | $1,31 \times 10^{-6}$  | $\text{m}^2\text{s}^{-1}$ |
| Densità CO <sub>2</sub>                     | 11,33                  | $\text{kg m}^{-3}$        |
| Tensione superficiale acqua-CO <sub>2</sub> | 0,045                  | $\text{N m}^{-1}$         |

**Tabella 4.2:** Proprietà fluide utilizzate nelle simulazioni finali di valvola e bottiglia

## 4.8 Regime fluido all'interno delle simulazioni di valvola e bottiglia

Il regime fluido può essere specificato all'interno del file `turbulenceProperties` contenuto all'interno della directory `constant`.

### 4.8.1 Simulazione finale della valvola

Le simulazioni finali sono state realizzate mediante tecnica DNS (Direct Numerical Simulation). Dunque, all'interno di `turbulenceProperties` si trova il comando

```
1 simulationType laminar;
```

che permette di non inserire nessun modello di turbolenza nella simulazione.

### 4.8.2 Simulazione finale della bottiglia

Le simulazioni finali per le tre portate sono state realizzate adottando il modello di turbolenza k- $\epsilon$  Standard. Dunque all'interno di `turbulenceProperties` si trovano le righe

```
1 simulationType RAS;
2
3 RAS
4 {
5     RASModel      kEpsilon;
6     turbulence    on;
7     printCoeffs  on;
8 }
```

L'impiego di un modello di turbolenza richiede condizioni iniziali per l'energia cinetica turbolenta e la velocità di dissipazione dell'energia cinetica turbolenta.

La condizione iniziale per  $k$  richiede l'uso dell'intensità della turbolenza. Infatti, la condizione iniziale si traduce in

$$k = \frac{3}{2}(I|U_{ref}^{\vec{}}|)^2 \quad (4.15)$$

dove  $I$  è l'intensità della turbolenza (posta 0,05) e  $U_{ref}^{\vec{}}$  è una velocità di riferimento, in questo caso posta pari alla velocità nella sezione di `inlet`, calcolata dividendo la portata liquida per la sezione di `inlet`.

Per quanto riguarda la condizione iniziale su  $\epsilon$ , si è utilizzata la stima

$$\epsilon = \frac{C_{\mu}^{0.75} k^{1.5}}{L} \quad (4.16)$$

dove  $C_{\mu}$  è una costante del modello pari a 0,09, mentre  $L$  è una lunghezza caratteristica del sistema. All'interno delle simulazioni la lunghezza caratteristica adottata è stata il diametro idraulico calcolato sulla sezione di `inlet`.

$$d_{idraulico} = \frac{d_{maggiore}^2 - d_{minore} \times d_{maggiore}}{d_{minore} + d_{maggiore}} \quad (4.17)$$

## 4.9 Controlli di simulazione

Essi vengono indicati all'interno del file `controlDict` all'interno della directory `system`. All'interno delle simulazioni è stato utilizzato un file `controlDict` del tipo

```

1 18 application      interFoam;
2 19
3 20 startFrom        startTime;
4 21
5 22 startTime         t_1;
6 23
7 24 stopAt           endTime;
8 25
9 26 endTime          t_2;
10 27
11 28 deltaT           0.00001;
12 29
13 30 writeControl      adjustableRunTime;
14 31
15 32 writeInterval     0.025;
16 33
17 34 purgeWrite        0;
18 35
```

```
19 36 writeFormat      ascii;
20 37
21 38 writePrecision  6;
22 39
23 40 writeCompression compressed;
24 41
25 42 timeFormat       general;
26 43
27 44 timePrecision    6;
28 45
29 46 runTimeModifiable yes;
30 47
31 48 adjustTimeStep   on;
32 49
33 50 maxCo             0.9;
34 51 maxAlphaCo       0.9;
35 52
36 53 maxDeltaT        0.001;
37 54
38 55
39 56 // ***** //
```

In esso è possibile specificare il solver utilizzato e il tempo di inizio e fine simulazione. Inoltre, alla voce `deltaT` si indica il passo temporale. Attivando il comando `adjustTimeStep` si abilita il codice a modificare il passo temporale a seconda del valore del numero di Courant, sul quale vengono imposte specifiche alla fine del file. In aggiunta, `writeInterval` specifica l'intervallo temporale in cui verranno stampate le cartelle contenenti i dati per i vari time step.

## Capitolo 5

# Creazione delle griglie e descrizione del film

### 5.1 Descrizione di una generica griglia computazionale in OpenFOAM

In questo capitolo viene descritto il metodo utilizzato in OpenFOAM per la creazione delle griglie computazionali. La mesh è parte integrale della soluzione numerica in quanto deve soddisfare alcuni criteri per assicurare l'ottenimento di una soluzione valida e accurata. Ogni volta che simulazione è in stato di `run`, OpenFOAM verifica che la griglia rispetti un set stringente di vincoli di validità e, nel caso ciò non si verifichi, la simulazione viene fermata.

Di default, OpenFOAM definisce una griglia formata da un numero arbitrario di celle poliedriche tridimensionali, delimitate da un numero arbitrario di facce poligonali, ovvero le celle possono avere un illimitato numero di facce dove, per ciascuna faccia, non vi è un limite di spigoli. Una mesh definita con questa struttura generale è nota in OpenFOAM come `polyMesh`. Questa definizione di mesh consente una grande libertà nella creazione e nella manipolazione delle griglie computazionali soprattutto nei casi in cui la geometria da discretizzare è complessa.

Come precedentemente espresso, in OpenFOAM la griglia computazionale viene definita dalla directory `polyMesh`. All'interno della directory `polyMesh` è possibile trovare i files di testo descrittivi in modo univoco le celle della griglia. Inoltre, fra questi files di testo è di particolare importanza il file `boundary`, il quale definisce e nomina le zone della griglia in cui verranno definite le condizioni al contorno.

## 5.2 Creazione di una mesh in OpenFOAM

Al fine di ottenere la griglia computazionale è dunque necessario ottenere la directory `polyMesh`, la quale verrà inserita all'interno della cartella base di simulazione nel percorso `case/constant`. All'interno del lavoro di tesi, per l'ottenimento della mesh si è fatto uso del tool `snappyHexMesh`.

La procedura di ottenimento della griglia computazionale si divide in vari step. In primo luogo è necessario creare una directory di base, nella quale siano presenti i percorsi `constant` e `system`. All'interno della directory `constant` è necessario creare le directory `triSurface`, dove dovranno essere inseriti i files `.stl` descrittivi della geometria da processare, e `polyMesh`, nella quale si provvederà alla creazione del file `blockMeshDict`. In secondo luogo, all'interno della directory `system` è opportuno che vi siano i files `surfaceFeatureExtractDict` e `snappyHexMeshDict`. Dopo aver creato tutto ciò, posizionandosi nella cartella base, la procedura si conclude digitando in sequenza i comandi `blockMesh`, `surfaceFeatureExtract`, `snappyHexMesh`. In questo modo la griglia viene ottenuta a partire da una geometria di riferimento.

Ulteriori dettagli sulla procedura verranno illustrati nei paragrafi successivi.

### 5.2.1 Geometria

In fluidodinamica computazionale, per geometria si intende la superficie geometrica che racchiude il campo di simulazione. Dunque, all'interno di questo lavoro di tesi, essa è rappresentata dal disegno CAD di valvola e bottiglia. Al fine di implementare la geometria in OpenFOAM, essa deve essere convertita in formato `.stl` (STereo Lithography). Un esempio di geometria della simulazione della bottiglia è riportata in Figura 5.1.

### 5.2.2 `blockMesh`

Il comando `blockMesh` richiama il file `blockMeshDict` e permette di creare un blocco di base, ovvero un parallelepipedo, che ingloba la geometria. Il blocco definisce il dominio computazionale suddiviso in blocchi esaedrici che definiscono il grado di refinement di base della griglia. L'esecuzione del comando `blockMesh` crea all'interno del percorso `constant/polyMesh` i files descrittivi la griglia cartesiana di base, la quale sarà soggetta ad elaborazione tramite i comandi successivi.

Come precedente espresso, digitando `blockMesh` si fa riferimento al file `blockMeshDict`. Esso è un file di testo contenente le indicazioni per la creazione del blocco di base. All'interno del `blockMeshDict` sono presenti diverse funzioni. La funzione `convertToMeters` permette di inserire un fattore di scala riguardante la definizione dei vertici del blocco. Successivamente, è presente la funzione `vertices`



**Figura 5.1:** Esempio di geometria in formato .stl

che consente di specificare, tramite coordinate cartesiane, gli otto vertici base del parallelepipedo. I vertici devono essere numerati secondo un ordine preciso da 0 a 7. La funzione `blocks` viene sviluppata mediante una dicitura del tipo

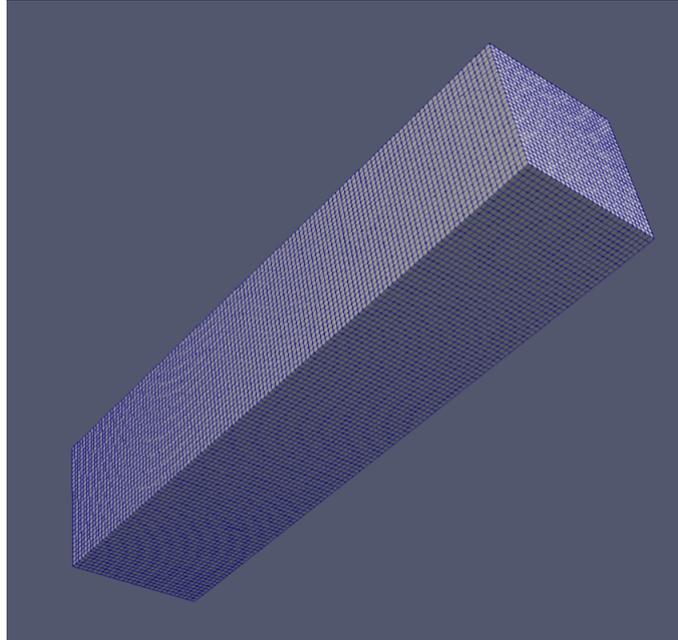
```

1 blocks
2 (
3 hex (0 1 2 3 4 5 6 7 ) (28 120 28 ) simpleGrading (1 1 1 )
4 ) ;

```

Ora, `blocks` descrive la suddivisione in celle del parallelepipedo di base. `hex` esprime che tutte le celle all'interno del blocco, definito dagli otto vertici, siano esaedriche. Successivamente, viene definito il numero di celle nelle tre direzioni, mentre con il comando `simpleGrading` è possibile definire il grado di espansione delle celle nelle tre direzioni. Da ciò emerge come è possibile calcolare la dimensione di una singola cella in una direzione semplicemente dividendo la dimensione della box in quella direzione per il numero di celle prescritto. Al fine di ottenere delle celle di base più uniformi possibile, è opportuno che in ogni direzione si mantenga un rapporto costante fra la dimensione della box e numero di celle. Proseguendo con la descrizione, la funzione `edges` definisce il tipo di spigolo che congiunge due vertici. Ci sono molte possibilità di scelta. Non inserendo nulla, come nel caso di questo lavoro, si sottintende l'utilizzo di una linea dritta (`line`) al fine di

congiungere due vertici. Successivamente, è possibile trovare le patches, ovvero delle regioni, il cui nome viene dato dall'utente. Infine, è possibile trovare il comando `mergePatchPair`, il quale non è stato preso in considerazione all'interno del lavoro. Un esempio di applicazione del comando `blockMesh` alla geometria presente in Figura 5.1 è presentato in Figura 5.2.



**Figura 5.2:** `blockMesh` applicato alla geometria della bottiglia

### 5.2.3 `surfaceFeatureExtract`

Il comando `surfaceFeatureExtract` richiama il file `surfaceFeatureExtractDict` e permette di creare, all'interno della directory `triSurface`, i file `*.eMesh` e, all'interno della directory `constant`, la cartella `extendedFeatureEdgeMesh`. In breve, l'obiettivo di questo comando è la creazione delle features, ovvero delle linee che verranno seguite dal tool `snappyHexMesh` al fine di realizzare griglie più uniformi possibile in presenza di spigoli.

### 5.2.4 `snappyHexMesh`

Il comando `snappyHexMesh` richiama il file `snappyHexMeshDict` e permette l'elaborazione del dominio creato con `blockMesh`. Il tool `snappyHexMesh` permette la creazione della griglia computazionale in tre fasi:

- **Castellated Mesh**, che crea una griglia solo con celle esaedrali e non perfettamente aderente gli \*.stl che definiscono la superficie. In questa prima fase viene effettuato un primo raffinamento delle celle contenute nella box di base. Per attivare questa fase, all'interno del file `snappyHexMeshDict` si pone la scritta `true` in corrispondenza della dicitura `castellatedMesh`, `false` altrimenti.
- **Snapping**, che prevede lo spostamento dei vertici sulla superficie \*.stl delle celle fuoriuscenti dalla geometria. Praticamente, si rende la griglia perfettamente aderente alla geometria di base. Per attivare questa fase, all'interno del file `snappyHexMeshDict` si pone la scritta `true` in corrispondenza della dicitura `snap`, `false` altrimenti.
- **Add Layers**, che permette l'aggiunta dei layers in prossimità della superficie di base. Per attivare questa fase, all'interno del file `snappyHexMeshDict` si pone la scritta `true` in corrispondenza della dicitura `addLayers`, `false` altrimenti.

All'interno del file `snappyHexMeshDict`, una delle prime sezioni è occupata dal comando `geometry`, all'interno del quale è opportuno definire tutte le superfici in corrispondenza delle quali la griglia verrà modellata ed infittita. Dunque, è necessario che siano inseriti tutti i files .stl definenti la geometria di base. Essi dovranno essere nominati mediante l'utilizzo del comando `triSurfaceMesh`. Inoltre, in questa sezione è anche possibile definire regioni del dominio fluido in cui si vuole effettuare un raffinamento della mesh. A questo proposito, di particolare importanza in questo lavoro di tesi, è stato il comando `searchableBox`, che permette di definire un parallelepipedo interno alla griglia, specificato mediante i due vertici estremi della diagonale.

Terminata la sezione `geometry`, si incontrano le sezioni:

- `castellatedMeshControls`, che governa la suddivisione e la rimozione di celle
- `snapControls`, che permette di adattare la griglia alla geometria di base
- `addLayersControls`, che permette di inserire strati di celle aderenti della superficie geometrica. Ogni strato creato viene aggiunto in direzione ortogonale rispetto alla geometria di base.

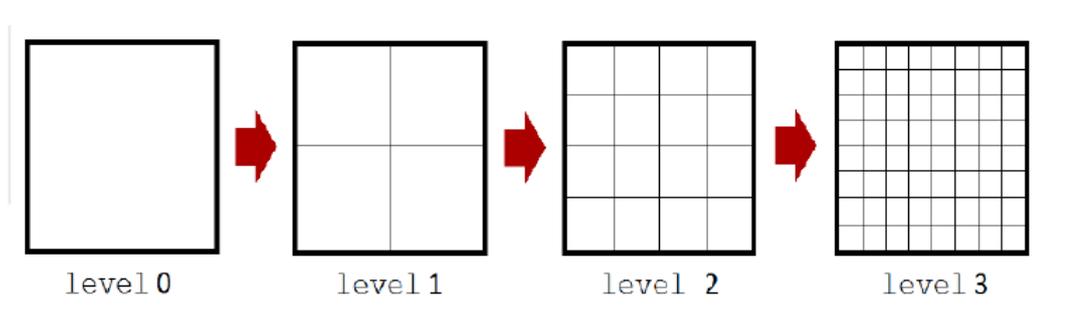
### **Castellated Mesh**

All'interno del file `castellatedMeshControls` si introducono le istruzioni volte all'identificazione delle celle da infittire e alla loro gestione numerica. Abilitando `castellatedMeshControls` viene avviata la procedura di `cellsplitting`. Nella prima sezione del dizionario vengono inseriti i parametri di rifinitura, che definiscono il numero di celle per processore, il limite di celle complessive, il numero di minimo

di celle per cui l'iterazione termina, il numero di strati in memoria tra livelli diversi. La procedura di rifinitura delle celle contenute nel blocco di base viene guidato dalle **features**, ovvero dalle superfici dove deve iniziare il processo di refinement, introdotte grazie ai files \*.eMesh. Per ciascuna **feature** viene esplicitato il livello di refinement della superficie in oggetto. Il livello di raffinamento  $n$  viene definito come

$$\Delta X_n = \frac{\Delta X_0}{2^n} \quad (5.1)$$

dove  $\Delta X_0$  indica la dimensione della cella di base, ovvero quella che si ottiene



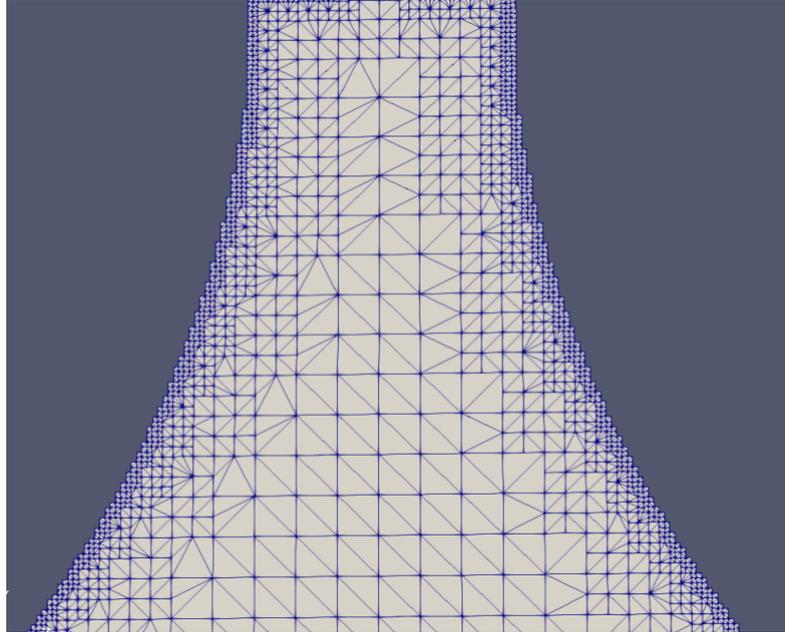
**Figura 5.3:** Diversi livelli di refinement a partire da una dimensione di cella base

tramite `blockMesh`. All'interno della Figura 5.3 sono mostrati i diversi livelli di refinement a partire da una dimensione di cella base. In aggiunta,  $\Delta X_n$  è dunque la dimensione della cella al livello  $n$ -esimo. All'interno del dizionario è poi possibile individuare la dicitura `refinementSurfaces`, che specifica il livello minimo e il livello massimo di raffinamento delle zone riguardanti gli \*.stl, ovvero al superficie geometrica. In aggiunta, il comando `refinementRegions` si applica alle box precedentemente definite all'interno della sezione `geometry` e permette, appunto, di definire il livello di refinement delle celle contenute all'interno di queste box. Inoltre, `refinementRegions` permette di definire per le box diversi modi di rifinitura:

- **inside**, ovvero vengono raffinate, al livello prescritto, le celle contenute all'interno della box
- **outside**, ovvero vengono raffinate, al livello prescritto, le celle contenute all'esterno della box
- **distance**, ovvero vengono raffinate, al livello prescritto, le celle situate ad una certa distanza prescritta dalla superficie della box.

Terminata la fase di `cellsplitting`, ovvero il processo di refinement delle celle, inizia il processo di rimozione delle celle. Vengono, ora, rimosse tutte le celle che

hanno almeno la metà del loro volume all'esterno della superficie geometrica. Un esempio di applicazione di `snappyHexMesh` con `castellatedMeshControls` attivato al blocco presente in Figura 5.2 è presentato in Figura 5.4.



**Figura 5.4:** Castellated Mesh in una porzione della bottiglia. Si noti la seghettatura presente lungo la superficie. Ciò è dato dalla non perfetta aderenza della mesh alla geometria di riferimento. Inoltre, viene effettuato un refinement delle celle del blocco di base

### Snapping

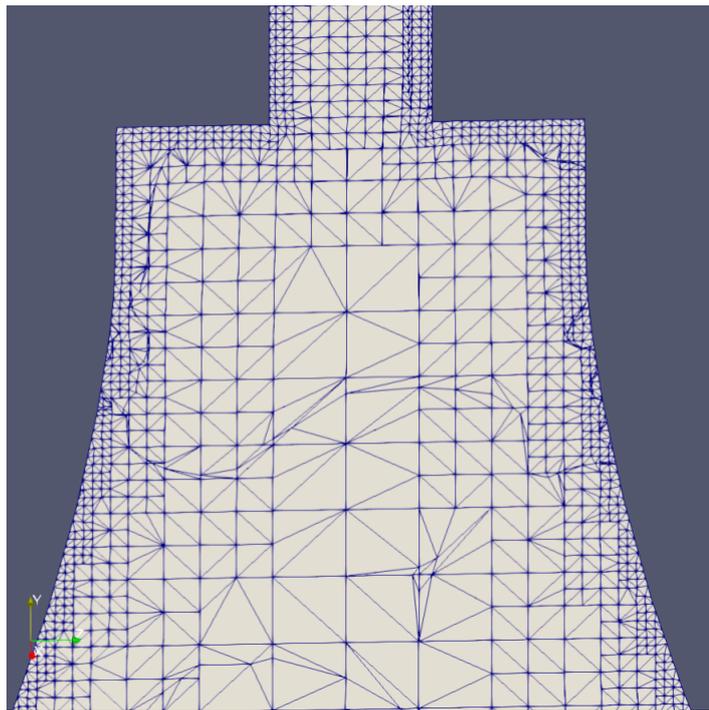
La mesh *castelled* risulta essere una griglia seghettata. Lo step di *snapping* consente di rendere la griglia *body-fitted* con la geometria. Il processo di *snapping* è iterativo ed è delineato dai seguenti passi:

- i vertici fuoriuscenti dalla geometria, appartenenti alle celle di confine, vengono spostati lungo la superficie geometrica definita dagli \*.stl;
- viene effettuato un controllo sulla qualità della mesh. Se essa non è soddisfatta si identificano i vertici che causano ciò mediante il loro spostamento;
- si riduce lo spostamento dei suddetti vertici dal punto iniziale e ci si assicura che i parametri di qualità della griglia siano soddisfatti. Il procedimento viene ripetuto fin tanto che la mesh non soddisfa i parametri di qualità.

All'interno del dizionario `snapControls` è possibile trovare i diversi comandi, fra i quali i più utilizzati risultano essere:

- `nSmoothPatch`, il cui valore più è grande e più si ottiene una superficie liscia;
- `tolerance`, che esprime la distanza alla quale `SnappyHexMesh` inizia la sua ricerca del punto in cui effettuare lo snap;
- `nSolveIter`, il quale valore regola il numero di iterazioni affinché si abbia lo spostamento della mesh;
- `nRelaxIter`, che rappresenta il numero massimo di iterazioni in cui `snappyHexMesh` compilerà lo script `relaxed` contenente i parametri di qualità della griglia non rispettati che permetteranno l'individuazione dei punti della mesh critici.

Per concludere, è possibile sottolineare come il processo di `snapping` sia il più oneroso in assoluto in termini di potenza computazionale richiesta e di tempo. Un esempio di applicazione di `snappyHexMesh` con `snapControls` attivato alla `Castellated Mesh` presente in Figura 5.4 è presentato in Figura 5.5.



**Figura 5.5:** Snapped Mesh in una porzione della bottiglia. Si noti come ora la mesh sia perfettamente aderente alla superficie della geometria.

## Add Layers

Questa sezione è di particolare importanza nel lavoro di tesi. Infatti, come verrà spiegato meglio in seguito, tramite l'inserimento di strati di celle ortogonali alla superficie di riferimento è possibile ottenere dalla simulazione una migliore risoluzione dello strato limite di film liquido aderente alla parete.

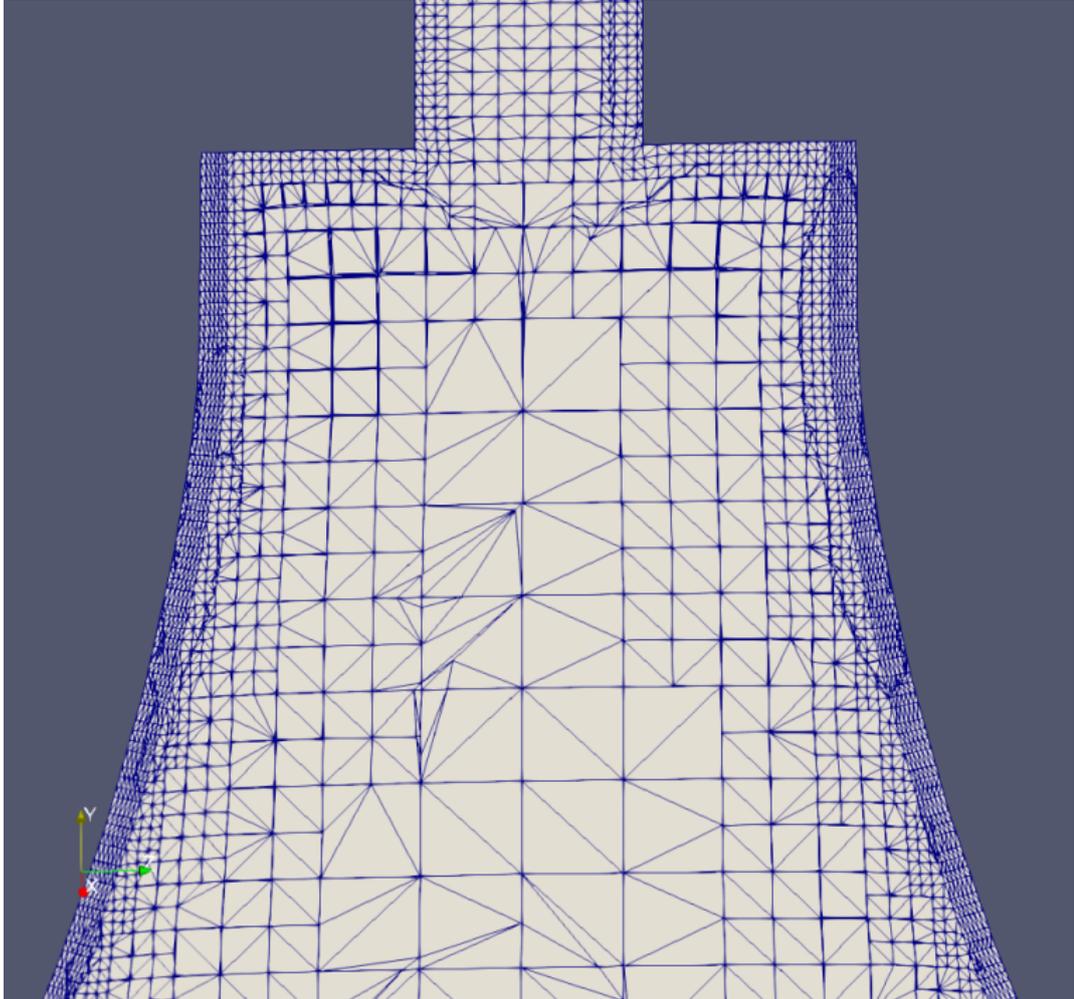
Il processo di aggiunta di un layer di celle prevede, in primo luogo, un allontanamento della griglia dalla superficie geometrica di uno spessore definito in `thickness`. Ora, viene verificato che la mesh rispetti i parametri di qualità e, in caso affermativo, il layer viene aggiunto. Segue poi un'ultimo check della griglia creata. Viceversa, in caso negativo, il layer non viene aggiunto, ma si riduce il suo spessore in un valore minore di quello prescritto in `thickness` in modo tale da avere meno spostamento della griglia e quindi avere più probabilità di soddisfare i parametri di qualità della griglia. Si andrà poi a verificare che la mesh sia soddisfacente e in caso affermativo si procede all'aggiunta dello strato di celle. In aggiunta, se i dettami di qualità della mesh non risultano essere mai rispettati, nessun layer verrà aggiunto.

La procedura di aggiunta dei layers è iterativa. Il primo layer è collocato in corrispondenza della superficie e ogni strato viene aggiunto dalla parte opposta alla superficie. L'aggiunta dei layers è governata dal dizionario `addLayersControls`. All'interno del dizionario `addLayersControls` è possibile incontrare il comando `relativeSizes`, al quale è possibile associare uno stato `true` o `false`. Con il termine `true` si prescrive che lo spessore dei layers sia definito in modo relativo alla dimensione delle celle adiacenti allo strato stesso. Con il termine `false`, invece, si indica lo spessore del layer in modo assoluto. Inoltre, il comando `expansionRatio` indica il rapporto di espansione fra lo spessore di un layer e il successivo. In aggiunta, `finalLayerThickness` indica lo spessore desiderato dello strato finale di celle, `firstLayerThickness` indica lo spessore desiderato dello strato iniziale di celle, mentre `minThickness` prescrive lo spessore minimo al di sotto del quale un layer non può essere aggiunto. Proseguendo nel dizionario, appare la sezione `layers`, all'interno della quale vengono inserite le superfici in corrispondenza delle quali vi deve essere l'inserimento dei layers. In conclusione, `nSurfaceLayers` indica il numero di starti da porre in corrispondenza della superficie.

Al fine di definire correttamente un boundary layer è obbligatorio definire una delle seguenti coppie di comandi all'interno di `addLayersControls`:

- `firstLayerThickness, thickness;`
- `firstLayerThickness, expansionRatio;`
- `finalLayerThickness, expansionRatio;`
- `finalLayerThickness, thickness;`
- `thickness, expansionRatio.`

Un esempio di applicazione di `snappyHexMesh` con `addLayers` attivato alla `Snapped Mesh` presente in Figura 5.5 è presentato in Figura 5.6.



**Figura 5.6:** Aggiunta di sei strati di celle alla parete superiore della bottiglia.

### 5.3 Descrizione del sottile film liquido alla parete: percorso di definizione della griglia finale

All'interno del capitolo precedente sono state presentate le simulazioni finali di valvola e bottiglia. Il percorso che ha portato alla costruzione delle griglie computazionali finali viene, ora, presentato in ordine logico. Nella prima sezione si

discute riguardo alla simulazione della valvola, mentre nella seconda sezione vengono illustrati i vari tentativi di griglia sulla bottiglia.

### 5.3.1 Griglie riguardanti la simulazione della valvola

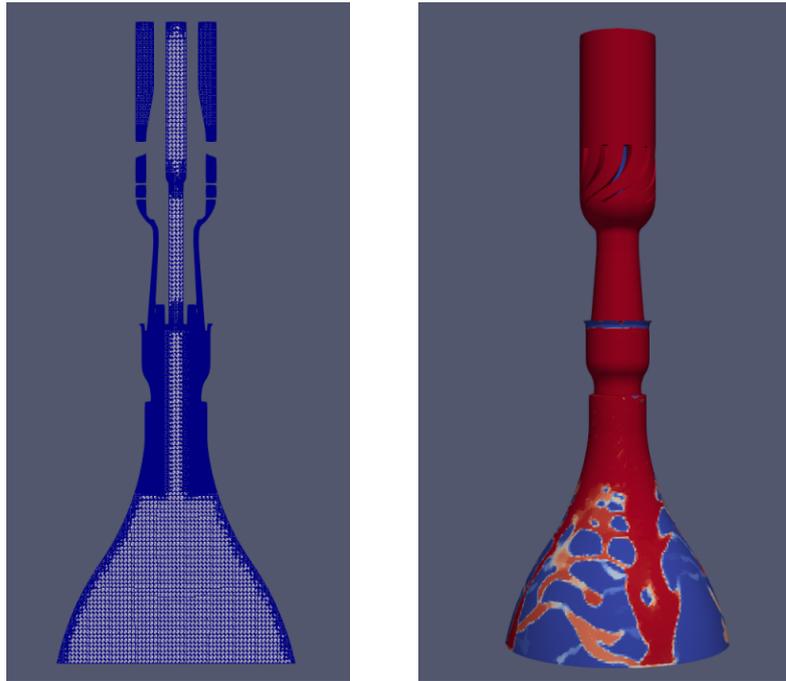
#### Griglia di prova

Una prima simulazione della valvola è stata condotta usando una griglia non ben definita e grossolana. Lo scopo di questa simulazione è stata la definizione e la creazione di una procedura tale per cui dai dati in output della simulazione della valvola si riuscissero ad ottenere i dati di input della simulazione della bottiglia. Questa prima fase di simulazione è stata eseguita per tutte e tre le portate  $Q_1$ ,  $Q_2$ ,  $Q_3$ . Le simulazioni sono state condotte senza l'utilizzo di un modello di turbolenza e per un tempo necessario a raggiungere le condizioni di stazionarietà.

Per quanto riguarda la discretizzazione del dominio, la griglia è stata raffinata in prossimità delle pareti e nella zona del tappo della bottiglia, come rappresentato in Figura 5.7. Infatti, esse sono le zone in cui è presente il film liquido. Nella Figura 5.7 è anche possibile vedere il profilo di frazione volumica alle pareti. In questa immagine si vede nettamente come vi sia un distacco del film liquido dalla parete appena superato il collo della bottiglia: veri e propri rivoli di fluido si vengono a formare. Nel caso di portata  $Q_1$ , inoltre, è stato osservato un anello di gas all'ingresso della bottiglia. Ciò non rappresenta bene la fisica osservata sperimentalmente.

#### Griglia finale

Questa seconda, e ultima, simulazione riguardante la valvola è stata realizzata utilizzando una griglia molto più raffinata e definita. Per quanto riguarda la griglia computazionale, essa presenta una zona raffinata molto più estesa rispetto alla precedente griglia, come è visibile in Figura 5.8. Ciò è stato fatto al fine di descrivere in modo più accurato i fenomeni d'interfaccia fra film liquido entrante e gas uscente dalla valvola. Utilizzando questa modifica all'interno della griglia non si sono risolti i problemi di scollamento del film liquido dalla parete della bottiglia. Si è comunque deciso di adottare questa simulazione della valvola come definitiva in quando il taglio, al fine di simulare la bottiglia, è stato effettuato nella zona d'imbocco della bottiglia, dove il film è ben aderente alla parete e quindi fisicamente accettabile.

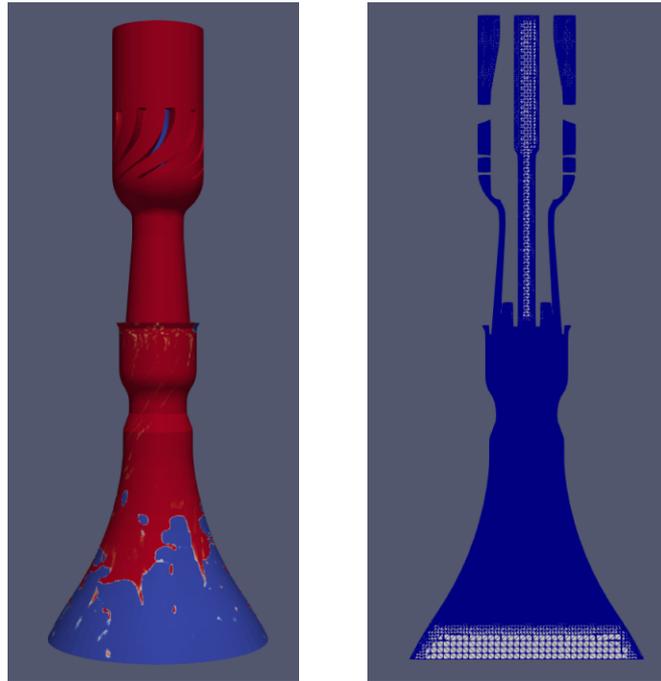


**Figura 5.7:** Sinistra: taglio della griglia computazionale di prova  
Destra: portata  $Q_3$ , alpha.water, time step 1 sec. Si noti lo scollamento del film liquido.

## 5.4 Griglie riguardanti la simulazione della bottiglia

### Problemi legati alle boundary condition

In questo primo tentativo di simulazione della bottiglia, si è considerato una taglio nella zona terminale della valvola. Dunque, si è adottata la geometria riportata in Figura 5.9. In questa fase si è utilizzata la griglia computazionale presente in Figura 5.9, avente sei milioni di celle. In essa, realizzata con `snappyHexMesh`, si può apprezzare l'infittimento nella zona finale della valvola, nella zona del tappo della bottiglia e lungo le pareti affinché si riesca ad ottenere una buona risoluzione del film liquido. Implementando la suddetta griglia nella simulazione delle tre portate, si è riscontrato un lungo tempo tempo di simulazione causato da un basso passo temporale, il quale è adattativo in modo tale da rispettare il numero di Courant critico. Analizzando le simulazioni si è ipotizzato che ciò fosse causato dal passaggio repentino delle boundary condition tra le sezioni di `inlet` e di `outlet`. Infatti, osservando i vettori velocità, emerge come nell'incontro fra le sezioni di `inlet` e



**Figura 5.8:** Sinistra: portata  $Q_3$ , alpha.water, time step 1sec. Si noti lo scollamento del film liquido.

Destra: taglio della griglia computazionale finale

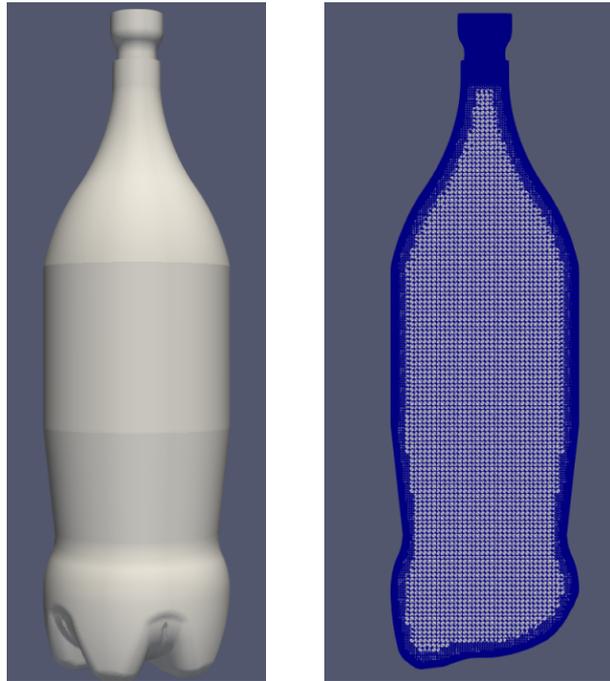
outlet si generino velocità enormi rispetto a quelle reali presenti all'interno della bottiglia. Ciò è ben visibile in Figura 5.10.

### Inserimento di una parete buffer

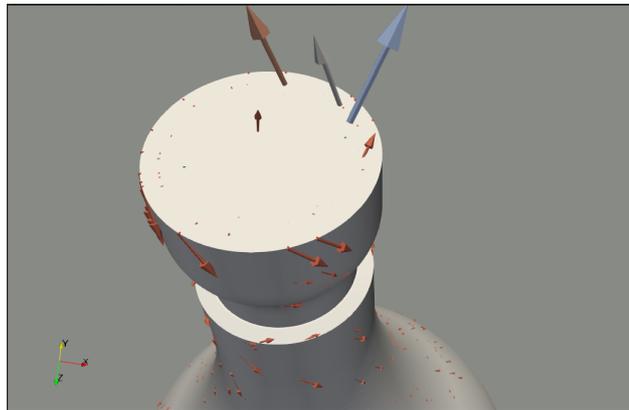
Lo step successivo è stato quello di introdurre uno spazio di buffer tra sezioni di inlet e di outlet. Esso è ben visibile all'interno della Figura 5.11. Questa sezione aggiuntiva fa, dunque, da intercapedine e viene trattata come wall. Con l'inserimento della sezione di buffer la simulazione è risultata essere più veloce negli istanti iniziali, raggiungendo però passi temporali dell'ordine di  $10^{-6}$  dopo appena pochi decimi di secondo. Il tentativo di accorciare i tempi di simulazione, dunque, non è andato a buon fine.

### Allontanamento delle sezioni di inlet ed outlet tramite una parete

Dopo l'insuccesso della modifica precedente, un altro tentativo è stato eseguito ponendo l'outlet a contatto con una wall, a sua volta a contatto con l'inlet. Tutto ciò è visibile in Figura 5.12. Si è deciso di eseguire ciò allontanando il

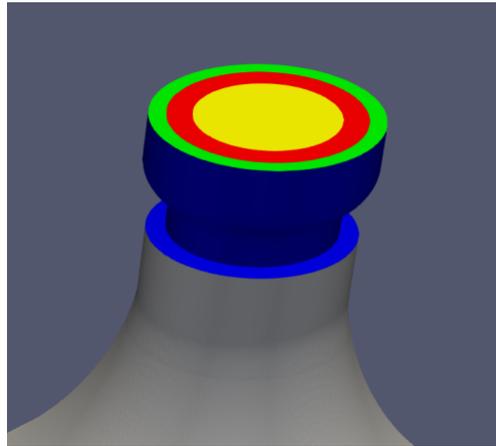


**Figura 5.9:** Sinistra: geometria implementata nel primo set di simulazioni. Si noti come la bottiglia sia legata ad una piccola porzione di valvola.  
Destra: griglia computazionale utilizzata all'interno del primo set di simulazioni. Si noti la presenza di una box infittita nella zona del tappo della bottiglia e l'infittimento generale alle pareti

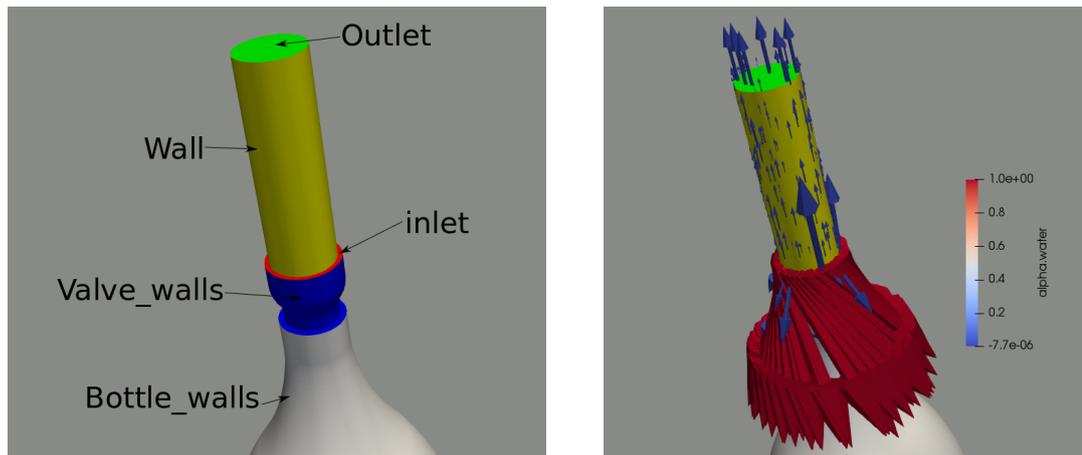


**Figura 5.10:** Vettori velocità enormi fra il confine delle sezioni di inlet ed outlet

più possibile outlet e inlet al fine di evitare conflitti fra le boundary condition. All'inlet si ha una swirl condition che imprime al flusso un moto rotazionale. La



**Figura 5.11:** Inserimento di una parete buffer. In giallo si indica l'outlet, ovvero la sezione di fuoriuscita del gas, in verde l'inlet, ovvero la sezione di ingresso della portata liquida e in rosso la parete buffer che separa le due sezioni.

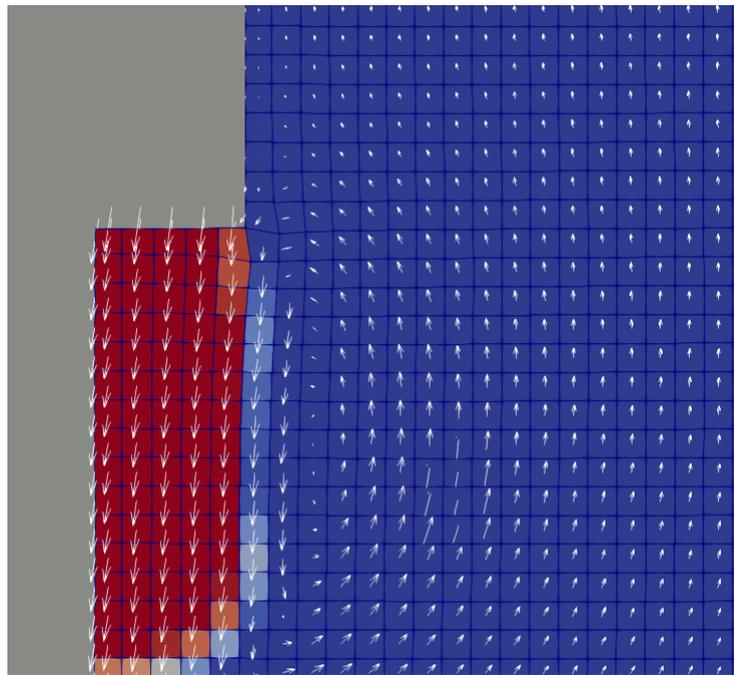


**Figura 5.12:** Sinistra: è possibile osservare la geometria adottata, nella quale vi è un allontanamento delle sezioni di inlet ed outlet.

Destra: vengono mostrati i vettori velocità derivanti da una delle simulazioni. Si noti come ora i vettori presenti al congiungimento delle sezioni di inlet ed outlet siano dello stesso ordine di grandezza per quanto riguarda il modulo dei vettori interni al campo di simulazione.

condizione al contorno idonea per la pressione sarebbe una pressione che anch'essa vari secondo una funzione radiale. Tuttavia è necessario imporre questa pressione: se in ingresso si ha una condizione di Dirichlet e Neumann per la pressione allora all'uscita è necessaria una condizione di Dirichlet in modo tale che il sistema

sappia a che pressione si trovi. Detto ciò, la condizione `totalPressure` forza il sistema ad avere la stessa pressione su tutta la superficie presa in esame. Ora, se si pone l'`outlet` vicino all'`inlet` ciò non è vero, e la simulazione è peggiore se le due sezioni sono a contatto. Allontanando l'`outlet`, il flusso ha tempo e spazio per raddrizzarsi e la pressione riesce ad avere una distribuzione tale per cui la condizione di `totalPressure` non disturbi il flusso. Con questi elementi introdotti, la simulazione ha delle velocità grandi al contatto tra `wall` e `inlet`, ma comunque risultano essere dello stesso ordine di grandezza. Dalla Figura 5.13 è possibile osservare i vettori velocità e la griglia abbastanza buona nelle zone di giunzione tra `inlet` e `outlet`. In questo modo anche i tempi di simulazione risultano essere



**Figura 5.13:** Nell'immagine viene mostrata la griglia computazionale nella zone di `inlet` e i vettori velocità. Inoltre è possibile osservare una certa diffusione del liquido all'interno della fase gassosa.

minori. La griglia in questione è stata realizzata tramite `snappyHexMesh` inserendo celle più grandi al bordo. Quest'ultima peculiarità accorcia di molto i tempi di simulazione. Nonostante ciò, il film liquido non rimane attaccato alla parete della bottiglia per molto e tende a formare veri e propri rivoli di fluido discendenti dalla parete della bottiglia contrariamente a quanto si vede sperimentalmente. Una causa di ciò può essere individuata nella zona di `inlet`. Infatti, come emerge dalla Figura 5.13, si crea una zona di diffusione di liquido verso la fase gassosa uscente dalla bottiglia. Ciò può provocare perdita di liquido all'interno del film e causarne

lo scollamento da parete.

### **Inserimento concomitante di una parete buffer e di una parete distanziatrice fra inlet ed outlet**

Visti i problemi legati alla diffusione del set di simulazioni precedente, affinché non si avesse deformazione del film liquido nella zona di **inlet**, vengono fatti tentativi diminuendo il diametro della colonna di fuoriuscita del gas e allontanandola con una parete buffer dalla zona di **inlet**. Ciò si evince confrontando la geometria in Figura 5.12 con quella presente in Figura 5.14. Inoltre, in questa parte vengono introdotte boundary condition per quanto riguarda gli angoli di contatto dovuti alla tensione superficiale tra liquido e i vari materiali che formano l'insieme valvola-bottiglia. Questa modifica permettere di non perdere informazioni sulla tensione superficiale ed avere una simulazione più accurata. Adottando questa geometria per le tre portate si nota come non vi sia più risalita del liquido verso la sezione di **outlet**. In queste simulazioni si osserva un distacco del flusso dalla parete all'ingresso della bottiglia, formando un vero e proprio anello di gas intrappolato nei primi decimi di simulazione. Esso, procedendo con i time step, tende a scomparire. Ciò è ben visibile all'interno della Figura 5.14.

Inoltre, si è notato la presenza di grosse bolle di gas, le quali tendono a risalire.



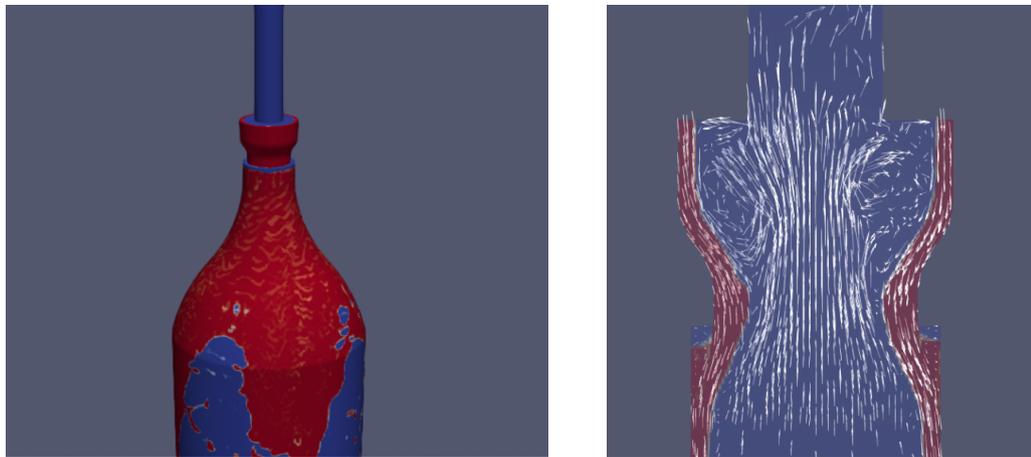
**Figura 5.14:** Sinistra: anello di gas intrappolato ad ingresso bottiglia nei primi istanti di simulazione.

Destra: scomparsa dell'anello di gas nei time step successivi e perfetta aderenza alla parete del film liquido ad ingresso bottiglia

Nonostante i tempi di simulazione ora siano accettabili, si riscontra comunque un distacco piuttosto rapido del film liquido dalla parete. Ciò è osservabile

all'interno della Figura 5.15. In questa fase, un'osservazione fatta è stata quella che l'idrofilia del materiale costituente la bottiglia condiziona il distacco fluido. Infatti, più il materiale è idrofilo e meno si ha distacco. All'interno delle simulazioni, però, vi è il vincolo del PET come materiale costituente le pareti della bottiglia. In aggiunta, all'interno della simulazione del riempimento, con questa geometria si sono osservati vortici di gas appena prima dell'ingresso bottiglia. Essi sono stati attribuiti ad uno sforzo di taglio esercitato dall'alto verso il basso da parte del film liquido di *inlet* e ad uno sforzo di taglio dal basso verso l'alto dovuto alla colonna d'aria ascendente e uscente dall'*outlet*. Questi vortici creano una sorta di effetto venturi creando una depressione localizzata. Ciò si evince dalla Figura 5.15.

In questa fase del lavoro di tesi ci si è concentrati anche allo sviluppo di una



**Figura 5.15:** Sinistra: si noti il precoce scollamento del film liquido alla parete della bottiglia.

Destra: vortici di gas nella porzione terminale della valvola.

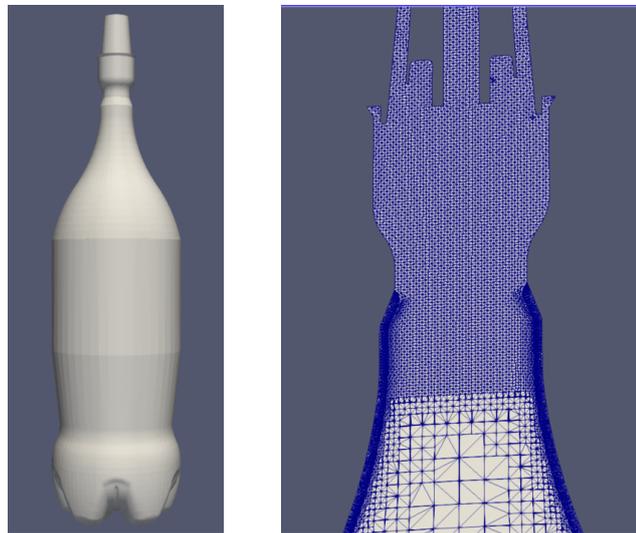
procedura, descritta in dettaglio all'interno del capitolo 6, di ottenimento della concentrazione della fase gassosa all'interno del battente liquido. Ora, simulando le tre portate per un certo periodo a portata costante e poi effettuando un riposo a portata nulla di 0,3 secondi per tutte e tre le portate, si è calcolato l'hold-up di gas nel battente liquido al termine della fase di riposo per le portate  $Q_1$  e  $Q_2$ . I valori calcolati sono riportati in Tabella 5.1. Essendo  $Q_2 > Q_1$  ci si aspetterebbe che il gas hold-up sia maggiore per la portata  $Q_2$ . Ciò è in disaccordo con i dati ottenuti e la causa è stata individuata nel non fisicità della simulazione, la quale produce un distacco del film liquido dalla parete che non avviene sperimentalmente.

| Portata | Gas Hold-Up |
|---------|-------------|
| $Q_1$   | 2,6%        |
| $Q_2$   | 2,2%        |

**Tabella 5.1:** Frazione di gas all'interno del battente liquido nel fondo della bottiglia dopo la fase di riposo per le portate  $Q_1$  e  $Q_2$ .

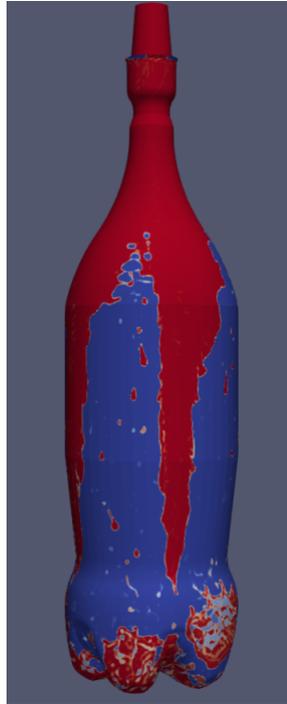
### Aggiunta di una porzione di valvola e adozione dei layers alle pareti

Ora, visti i risultati precedenti, ci si è focalizzati sul descrivere accuratamente il film liquido utilizzando strati di celle alla parete, ovvero i layers. Essi possono essere implementati mediante `snappyHexMesh` abilitando l'opzione `add Layers` in `snappyHexMeshDict`, come accuratamente descritto nella sezione 5.2.4. Inoltre, in questa fase della tesi si è considerata una porzione aggiuntiva di valvola, come si può vedere in Figura 5.16, permettendo così di non avere eccessive approssimazioni sulle condizioni di ingresso di liquido nel sistema. Ovviamente, per realizzare ciò, all'interno della simulazione della valvola è stata effettuato un taglio alla medesima altezza a cui si vuole inserire il liquido all'interno della simulazione della bottiglia. Ora, si sono realizzate simulazioni implementando 10 strati di celle nella parte superiore della bottiglia e 5 strati di celle nella parte inferiore della bottiglia. Ciascun strato di celle è spesso 0,0002 m. In Figura 5.16 è possibile osservare un taglio della griglia computazionale implementata nella zona superiore. Nonostante queste



**Figura 5.16:** Sinistra: geometria contenente una porzione maggiore di valvola. Destra: taglio di parte della griglia computazionale. Si noti la presenza di strati di celle alle pareti della bottiglia e l'infittimento generale nella zona della valvola.

modifiche, si è sempre osservato il distacco del film liquido dalla parete della bottiglia come è visibile in Figura 5.17.



**Figura 5.17:** Visibile distacco del film liquido anche con la nuova geometria e griglia adottate.

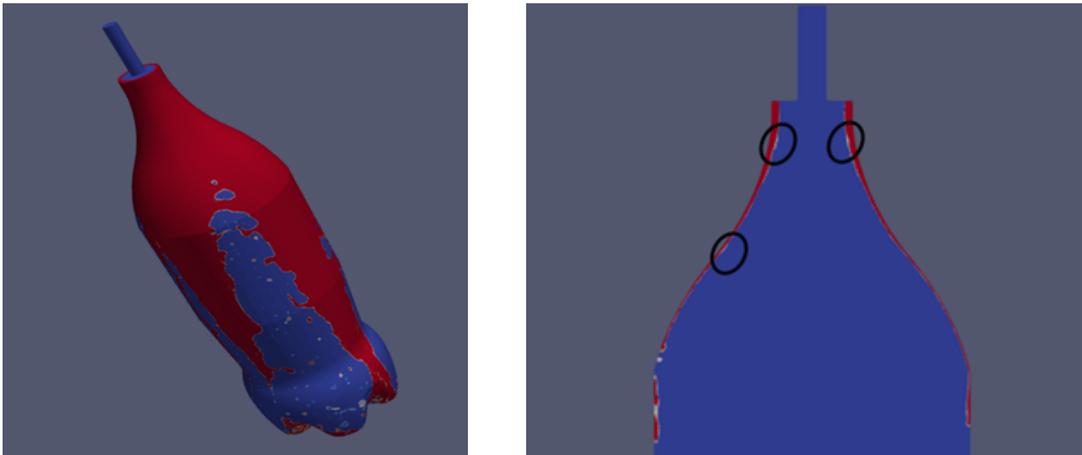
**Adozione del vecchio tipo di geometria costruita su nuovi files .stl di base e dati di inlet ottenuti dalla simulazione terminale della valvola**

Dai tentativi precedenti, allontanando le sezioni di `inlet` e di `outlet` si è risolto il problema della diminuzione esponenziale del time step dovuto all'iterazione fra le boundary condition. Ora, non avendo risolto il problema del distacco del film liquido a parete con i tentativi precedenti, si decide nuovamente di inserire il film liquido direttamente in bottiglia e di descriverlo con strati di celle facendo attenzione ad intrappolare bene la zona d'interfaccia.

In questa fase, dunque, si sono inseriti 9 layers alla parete superiore della bottiglia e 4 layers alla parete inferiore della bottiglia. Inoltre, è stata raffinata la zona del tappo. Tutto ciò è visibile in Figura 5.19, dove viene riportato un taglio della griglia computazionale nella zona superiore,

Analizzando la simulazione, si noti come si abbia meno distacco rispetto al tentativo precedente, osservabile in Figura 5.18.

In aggiunta, si sottolinea come nei punti evidenziati in Figura 5.18 vi sia un'evidente diffusione dell'interfaccia, dunque essa potrebbe ancora risultare poco raffinata. Notando ciò si è posta l'attenzione ad inglobare completamente l'interfaccia all'interno del boundary layer al fine di descrivere il più accuratamente possibile le zone di diffusione. Il fenomeno è complesso e va utilizzata una griglia fitta. Si infittisce nelle zone di diffusione perché si hanno gradienti alti e quindi con griglie troppo lasche si potrebbero avere alti errori numerici. Inoltre nelle zone di diffusione si perde massa, ovvero si ha che parte del liquido si stacca dall'interfaccia trasferendosi nella fase gassosa e provocando minore adesione del liquido a parete.

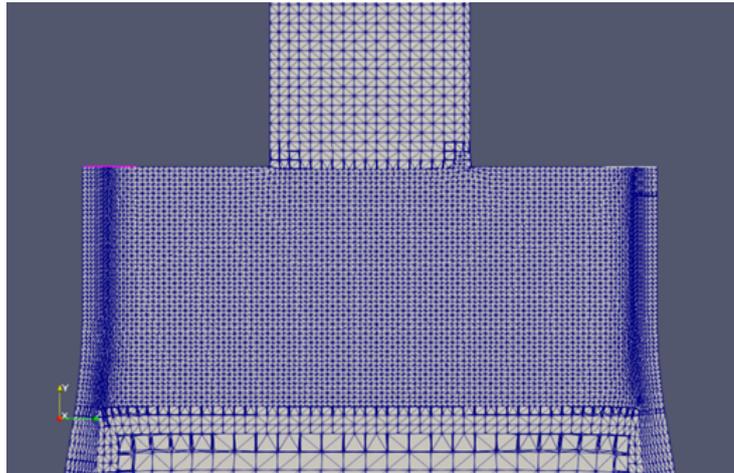


**Figura 5.18:** Sinistra: evidente scollamento del film liquido a parete, comunque minore rispetto a quello ottenuto nel set di simulazioni precedente. Destra: nel taglio vengono evidenziate varie zone di diffusione del film liquido all'interfaccia.

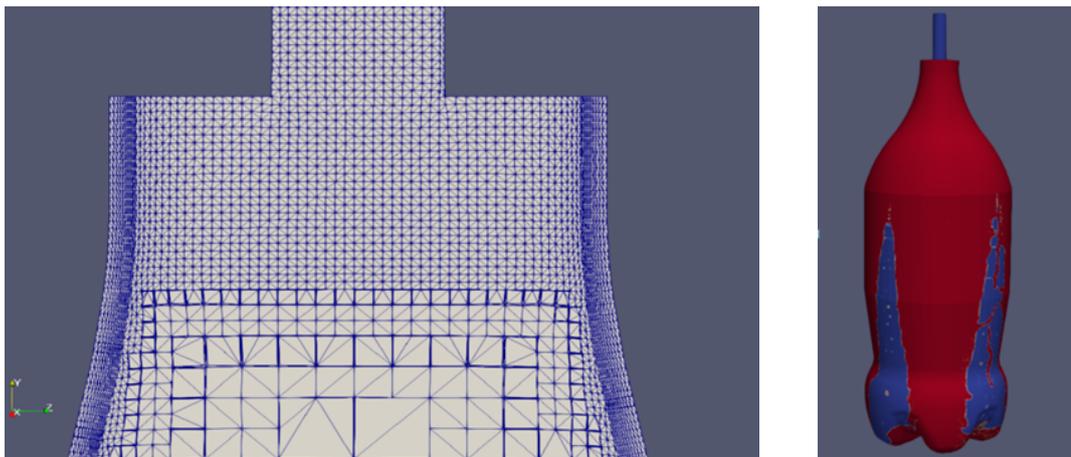
### **Aggiunta di strati di layers e deraffinamento nella zona del tappo**

Visti i risultati prodotti dal precedente set di simulazioni, si decide di adottare 10 layers nella parete superiore della bottiglia e 5 nella parete inferiore. Inoltre, notando la deformazione della griglia precedente a causa dell'elevato livello di raffinamento nella zona del tappo, si decide di diminuire il livello di raffinamento in questa zona. L'adozione di queste scelte permette di ottenere una griglia più uniforme, come è visibile in Figura 5.20.

Come si può osservare in Figura 5.20, la descrizione del film liquido è leggermente migliore rispetto ai tentativi precedenti. Da questo set di simulazioni si conclude che il metodo VOF è altamente sensibile alla deformazione della griglia. Nel successivo tentativo, infatti, si cercherà ancora di diminuire la deformazione della griglia dovuta alla compressione dei layers a seguito della loro aggiunta.



**Figura 5.19:** Taglio della griglia computazionale nella zona superiore. Si noti come la presenza di una box raffinata e dei layers deformi la griglia e la vada a comprimere. Inoltre, con il trattino viola si indica il diametro della sezione di *inlet*: si evidenzia dunque la volontà di inglobare il film liquido all' interno della porzione raffinata.



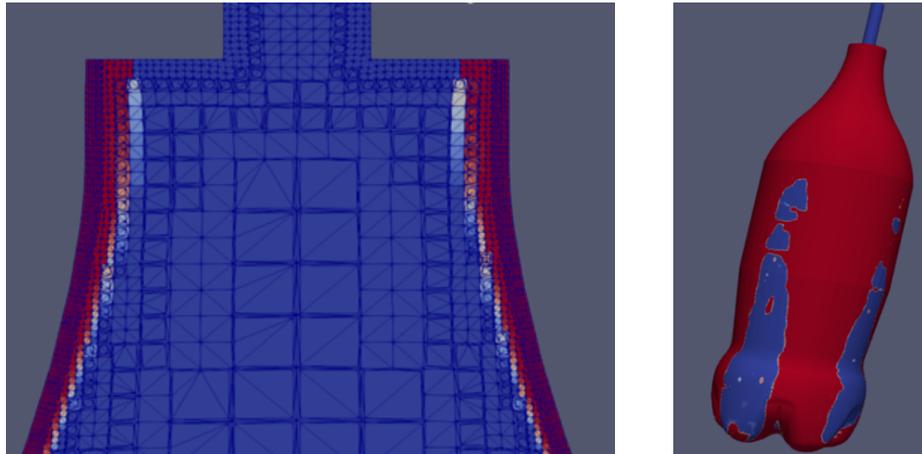
**Figura 5.20:** Sinistra: taglio della griglia computazionale nella zona superiore. Si noti come essa è molto meno deformata rispetto al tentativo precedente. Destra: lo scollamento del film liquido è ancora presente. Si ha, però, un progressivo miglioramento.

### **Eliminazione della box raffinata nella zona del tappo e diminuzione dei layers**

In questa fase, al fine di rendere la griglia ancora più uniforme, si è deciso di tentare un set di simulazioni senza box di raffinamento nella zona del tappo della bottiglia

e di diminuire lo strato di layers, ora 6 nella parte superiore della bottiglia e 6 nella parte inferiore della bottiglia. Da queste simulazioni si ottiene che il film si stacca sempre. Inoltre si osserva come 6 layers siano troppo pochi: infatti, non si riesce a descrivere bene l'interfaccia e ad analizzare bene la zona diffusiva. In aggiunta, si nota la presenza di enormi bolle alle pareti.

La griglia utilizzata e il distacco del film liquido sono osservabili in Figura 5.21.



**Figura 5.21:** Sinistra: taglio della griglia computazionale nella zona superiore. Si noti come 6 layers e l'assenza di infittimento non permettano una corretta descrizione del fenomeno diffusivo d'interfaccia.

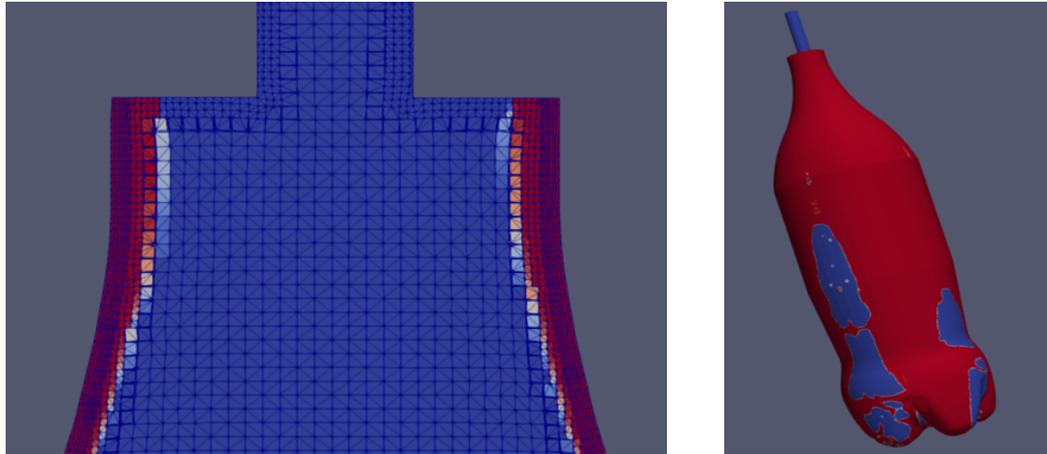
Destra: scollamento del film liquido dalla parete nonostante le modifiche.

### **Reintroduzione della zona raffinata e mantenimento del numero di layers al fine di mantenere sempre la griglia uniforme.**

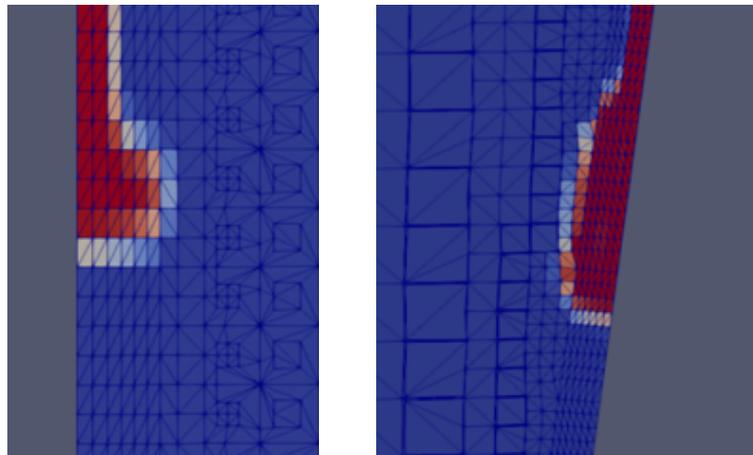
Avendo notato come uno scarso raffinamento della zona del tappo non riesca a descrivere bene le zone diffusive, ora, si decide di raffinare nuovamente e di non aumentare il numero di layers per evitare una compressione della griglia e renderla più uniforme possibile. La parte della griglia computazionale è visibile all'interno della Figura 5.22. Il risultato è migliore, non c'è la formazione di bolle alla parete. Si hanno solo due distaccamenti del film liquido, come si può vedere all'interno della Figura 5.22.

Visto il netto miglioramento, si è analizzato uno dei punti di distacco dell'interfaccia. I punti di distacco sono visibili all'interno della Figura 5.23. Si vede che la zona diffusiva non viene intrappolata benissimo all'interno dei layers. Il prossimo passo è stato dunque aumentare il numero di layers. Infatti, nelle altre zone dove non si ha distacco del film si ha che il film è sempre contenuto nel boundary layers.

Visto che aggiungendo 6 layers non si ha deformazione eccessiva della griglia, nel successivo tentativo vengono aggiunti altri 2 layers. Dunque, si avrenno 8 layers sia nella parete superiore che inferiore della bottiglia.



**Figura 5.22:** Sinistra: taglio della griglia computazionale nella parte superiore. Destra: notevole miglioramento riguardante il problema dello scollamento del film liquido.

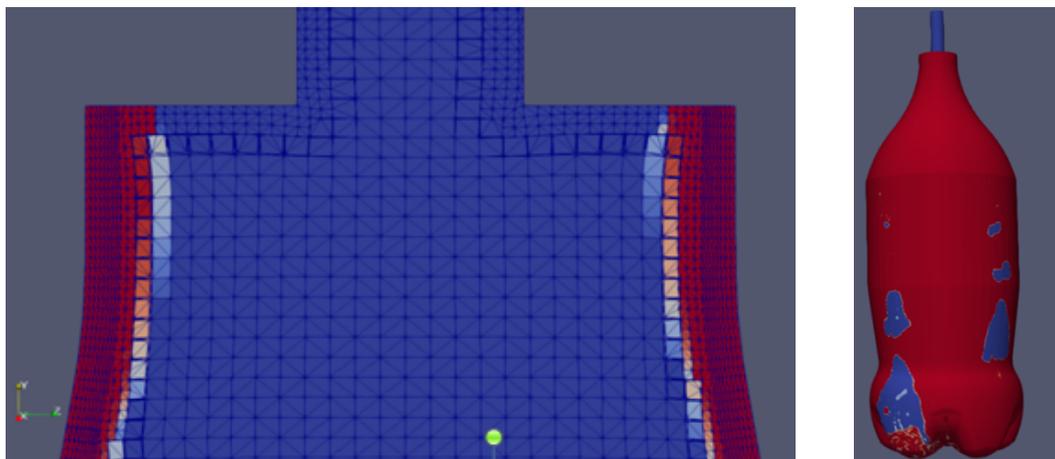


**Figura 5.23:** Punti di distacco dell'interfaccia. Si vede come il distaccamento avvenga dove il film non è completamente contenuto nello strato di layers.

### Aumento del numero di layers

Ora, avendo portato il numero dei layers a 8 in tutta la bottiglia, si nota che nella parte inferiore della bottiglia i layers intrappolano bene le zone diffusive, ma si hanno sempre dei punti di distaccamento. Si nota anche qui un notevole miglioramento: si hanno solo due distaccamenti e il film è uniforme in quasi tutte le zone. Per quanto riguarda la zona del tappo, le zone diffusive non vengono intrappolate bene e quindi è probabile che possa essere l'unica causa di questi distaccamenti.

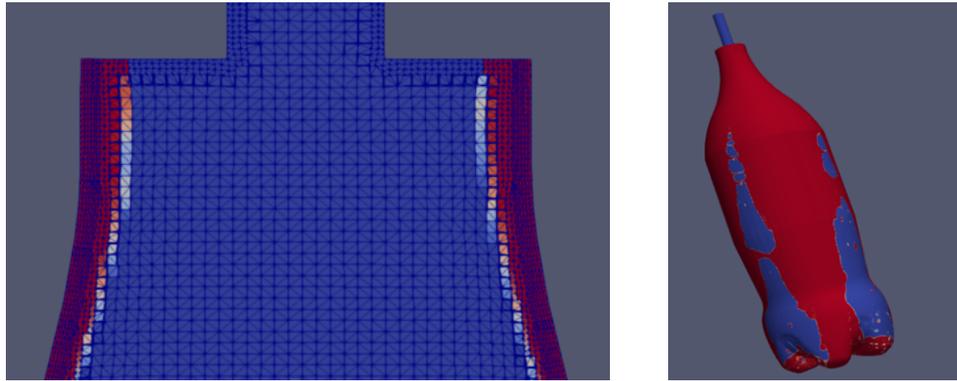
Una porzione della griglia computazionale e il notevole miglioramento nella descrizione del film liquido sono visibili all'interno della Figura 5.24.



**Figura 5.24:** Sinistra: taglio della griglia computazionale nella parte superiore. Si noti come le zone diffusive non vengano descritte bene. Destra: ulteriore miglioramento riguardante il problema dello scollamento del film liquido.

### Diminuzione della dimensione delle celle costituenti il blocco di base

Visto il tentativo precedente e data la volontà di mantenere il più possibile uniforme la griglia, si utilizzano 6 layers sia nella parte superiore che inferiore della bottiglia, ma si diminuisce del 20% la dimensione delle celle di base del `blockMesh`. Ciò perché è probabile che il calcolo sia influenzato anche dalla dimensione delle celle lungo la lunghezza della circonferenza e non dipenda solo dalla dimensione dei layers. Questo tentativo va nettamente peggio, come emerge dalla Figura 5.25. Le celle troppo piccole di base con l'aggiunta dei layers deformano la griglia, e ciò è possibile osservarlo in Figura 5.25. Da ciò si ribadisce come sia importante mantenere una griglia il più uniforme possibile nel metodo VOF.



**Figura 5.25:** Sinistra: taglio della griglia computazionale nella parte superiore. Si noti come vi sia una certa deformazione della griglia.

Destra: importante peggioramento riguardante il problema dello scollamento del film liquido. Diventa sempre più palese l'importanza di avere una griglia uniforme

### **Soluzione al problema del distacco del film liquido dalla parete: adozione di un modello di turbolenza**

Dati i vari tentativi di griglia si è constatato che il problema della descrizione del film liquido potesse non dipendere solo dalla griglia utilizzata. Infatti, dal calcolo del "numero di Reynolds a parete" emerge come all'interno delle simulazioni della bottiglia non ci si trovi in regime laminare, ma turbolento.

Ora, definendo il numero di Reynolds a parete come

$$Re_{parete} = \frac{4\rho U_y s}{\mu} \quad (5.2)$$

dove  $s$  è lo spessore del film liquido, emerge come esso sia in pieno campo turbolento per le tre portate simulate. Il confine fra regime laminare e turbolento si colloca ad un valore di Reynolds a parete pari a 1500. Da ciò emerge come sia opportuno adottare un modello di turbolenza al fine di descrivere correttamente il film liquido alla parete.

All'interno delle simulazioni è stato adottato il modello  $k-\epsilon$ , come descritto all'interno del capitolo 4, e ciò ha permesso di ottenere una simulazione coerente con l'evidenza sperimentale per tutte e tre le portate, come verrà mostrato nel capitolo successivo. Riassumendo, per quando riguarda la bottiglia, non è stato possibile descrivere la turbolenza in modo diretto perché sarebbero state necessarie griglie di calcolo troppo fitte. Data la novità dell'applicazione e data l'alta sensibilità del metodo VOF nei confronti della griglia computazionale utilizzata, si è cercato di capire se fosse possibile descrivere bene la fisica del riempimento mediante diversi tentativi di griglia. All'interno del lavoro di questa tesi, dunque, stati realizzati diversi tentativi

di griglia, finalizzati alla descrizione del film liquido alla parete della bottiglia con diversi strati di celle ortogonali alla parete di spessore millimetrico denominati layers. Nonostante questi tentativi di modifica della griglia computazionale finalizzati ad ottenere un film liquido continuo alla parete della bottiglia durante il riempimento della bibita, infine, il problema del distacco del film liquido dalla parete della bottiglia è stato risolto adottando un modello di turbolenza.

# Capitolo 6

## Risultati

### 6.1 Determinazione angolo di swirl e velocità tangenziale

Come già illustrato all'interno del capitolo 4, si simula la valvola più una porzione di bottiglia in modo tale da ricavare le condizioni di `inlet` della simulazione della bottiglia, le quali sono essenzialmente la velocità rotazionale, espressa in rpm, del film liquido attaccato alla parete.

All'interno del lavoro di tesi è stata sviluppata una procedura che permette per ogni portata simulata il calcolo dell'angolo di swirl e della velocità tangenziale del film liquido aderente alle pareti. Questa procedura è applicabile in una qualsiasi sezione di taglio ortogonale all'asse verticale ed è stata realizzata tramite `Paraview`. Da questi due parametri, tramite foglio di calcolo, è possibile ricavare la corrispondente velocità rotazionale. La procedura creata è la seguente:

- si consideri un time step tale per cui ci si trovi allo stazionario. Si selezioni poi la vista `alpha.water`, che permette di visualizzare la distribuzione della frazione volumica all'interno del dominio fluido;
- tramite il comando `Slice` si esegue un taglio ortogonalmente all'asse verticale nella sezione in cui si vogliono determinare la velocità tangenziale e l'angolo di swirl del film liquido. All'interno della Figura 6.2 è possibile osservare il posizionamento del comando `Slice` affinché si ottenga la sezione di taglio;
- all'interno di queste sezioni si isola il film liquido, ovvero si considera la zona con  $\alpha$  prossimo ad 1. Ciò è ottenibile mediante il comando `Threshold` applicato ad `alpha.water`, in cui è stato impostato come limite minimo 0,95 e limite massimo 1. In questo modo, all'interno della sezione tagliata si considerano solo le zone in cui  $\alpha$  è compreso fra 0,95 ed 1. Si ricordi che la

sezione individuata è sul piano X-Z in quanto la valvola ha come asse verticale un asse parallelo all'asse Y;

- ora, si applica un filtro **Calculator** impostandolo in modalità **Cell Data**. Tramite la formula

$$Tan\_Velocity = \sqrt{U_x^2 + U_z^2} \quad (6.1)$$

viene calcolata la velocità tangenziale in ogni punto del film liquido;

- proseguendo, si applica un secondo filtro **Calculator** impostandolo in modalità **Cell Data**. Tramite la formula

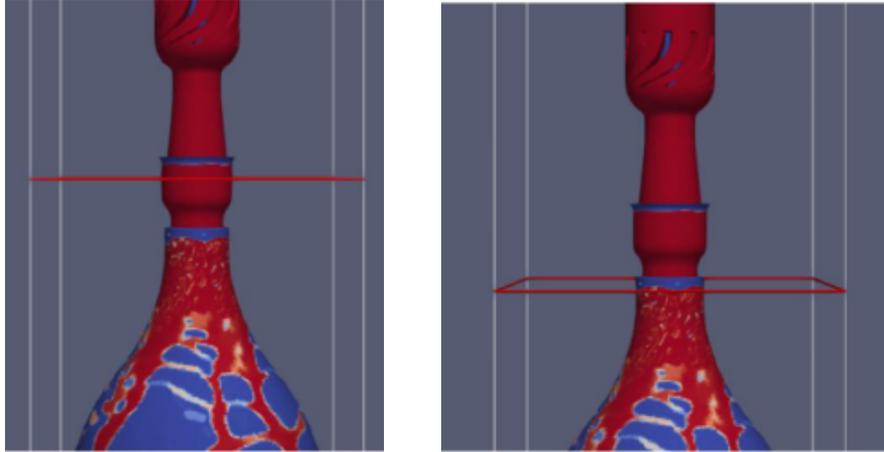
$$\arctan\left(\frac{|U_y|}{Tan\_Velocity}\right) \times \frac{180}{\pi} \quad (6.2)$$

viene calcolato l'angolo di swirl (°) su ogni punto del film liquido;

- si esegue una media di queste due grandezze su tutto il film liquido. Ciò è ottenibile mediante il comando **Integrate Variables**, spuntando **divide by cell Volume** in modo da ottenere la media della grandezza voluta;
- La procedura si conclude con il calcolo degli rpm. Viene implementata la formula

$$rpm = \frac{Tan\_Velocity \times 60}{Raggio\_Film \times 2 \times \pi} \quad (6.3)$$

dove *Raggio\_Film* è il raggio medio (m) del film liquido, facilmente leggibile da **Paraview**.



**Figura 6.1:** Portata  $Q_1$ , alpha.water, time step 1 secondo: due diversi posizionamenti del comando **Slice** affinché si ottenga la sezione di taglio voluta.

## 6.2 Gas intrappolato all'interno del battente liquido: procedura e risultati

Tramite **Paraview** è stata creata una procedura tale per cui è stato possibile, per ogni portata di riempimento simulata, calcolare il gas intrappolato in ogni istante temporale all'interno del battente liquido presente nel fondo della bottiglia.

L'idea di base è stata che il gas contenuto all'interno del liquido è ad una pressione maggiore di quella del gas sovrastante (pressione di riempimento  $P_{tank}$ ) a causa del battente idraulico. Visualizzando il campo di frazione volumica per ogni time step tramite **Paraview**, si è individuato un valore minimo di `p_rgh` tale per cui si riuscisse, per ogni time step, ad intrappolare bene il battente liquido.

### 6.2.1 Procedura per il calcolo del gas hold-up tramite **Paraview**

La procedura creata è la seguente:

- mediante il comando **Threshold** si inseriscano il valore minimo e il valore massimo riferiti a `p_rgh` tali per cui ad ogni time step si riesca ad intrappolare il volume di interesse. Questi valori minimo e massimo di `p_rgh` dipendono solo dalla pressione a cui viene posta la bottiglia durante la fase di riempimento della bibita gassata e, dunque, non variano per le portate simulate. L'applicazione di questo comando definisce un volume all'interno del dominio fluido;
- proseguendo, si applica un filtro **Integrate Variables** spuntando **divide by cell Volume**. Esso permette di integrare la frazione volumica sul volume ottenuto precedentemente. Da ciò ottengo la frazione volumica di liquido contenuta nel volume;
- affinché si ottenga il valore della frazione di gas contenuto nel volume, si applica un filtro **Calculator**. Esso, mediante la formula

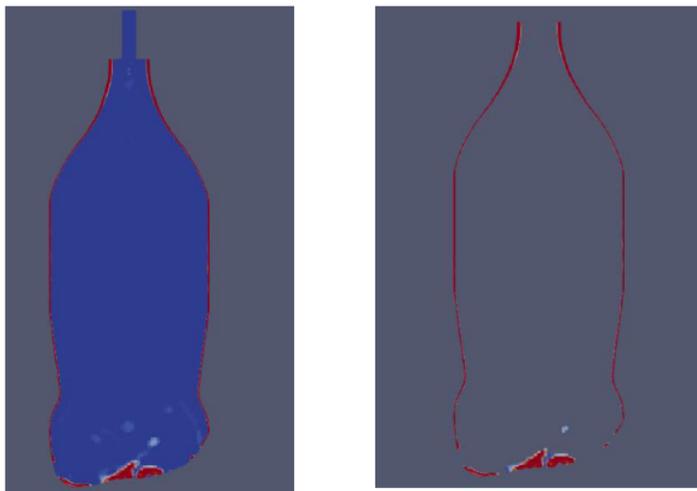
$$Gas\_Hold\_Up = 1 - \text{alpha.water} \quad (6.4)$$

permette di individuare la grandezza richiesta;

- i tre passi precedenti devono essere applicati ad ogni time step simulato. In questo modo si otterrà un valore di gas hold-up per ogni time step. Inserendo, poi, in ordinata i valori di gas hold-up e in ascissa il time step si ottiene l'andamento del gas intrappolato nel battente liquido nei confronti del tempo. Per fare ciò si applica il filtro **PlotDataOverTime**.

La bontà di questa procedura può essere mostrata dalle seguenti immagini, nelle quali sono presenti vari time step in cui viene mostrato l'effetto del **Threshold** applicato.

All'interno delle Figure 6.2, 6.3, 6.4 sono presentati, quindi, diversi istanti temporali all'interno delle simulazioni. Si vede chiaramente come la procedura effettuata abbia un ottimo intrappolamento del battente liquido ad ogni istante temporale di simulazione.



**Figura 6.2:** Portata  $Q_1$ , `alpha.water`, time step 0,5 secondi.

Sinistra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo pre-Threshold.

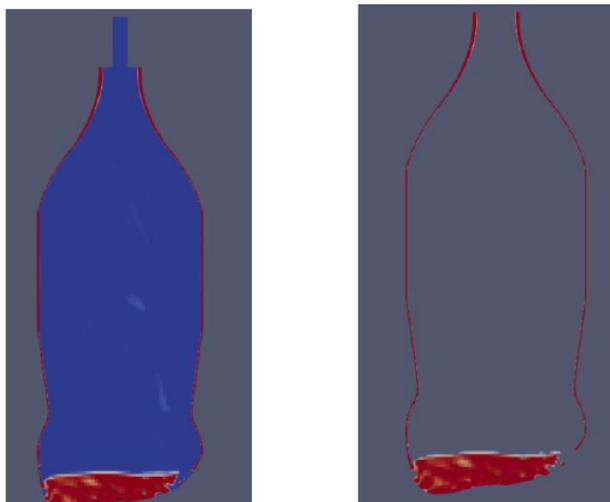
Destra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo post-Threshold.

### 6.2.2 Gas hold-up: risultati

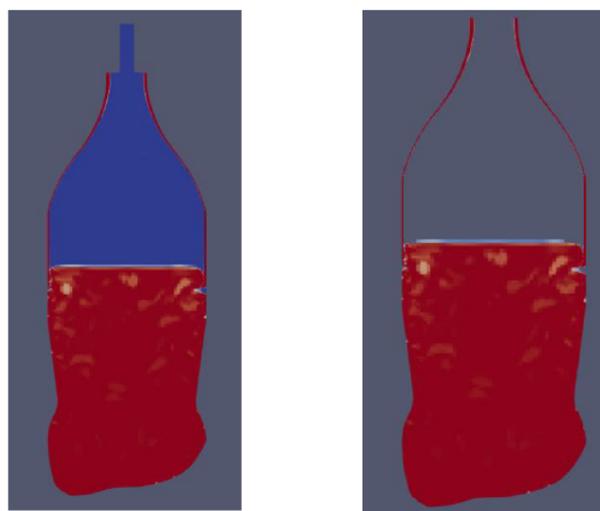
La procedura precedentemente descritta è stata applicata alle simulazioni finali della bottiglia per le portate di riempimento  $Q_1, Q_2$  e  $Q_3$ .

Da ciò si sono ottenuti i seguenti grafici: l'andamento della frazione di gas all'interno del battente liquido per la portata  $Q_1$  è visibile all'interno della Figura 6.5, per la portata  $Q_2$  all'interno della Figura 6.6 e per la portata  $Q_3$  all'interno della Figura 6.7.

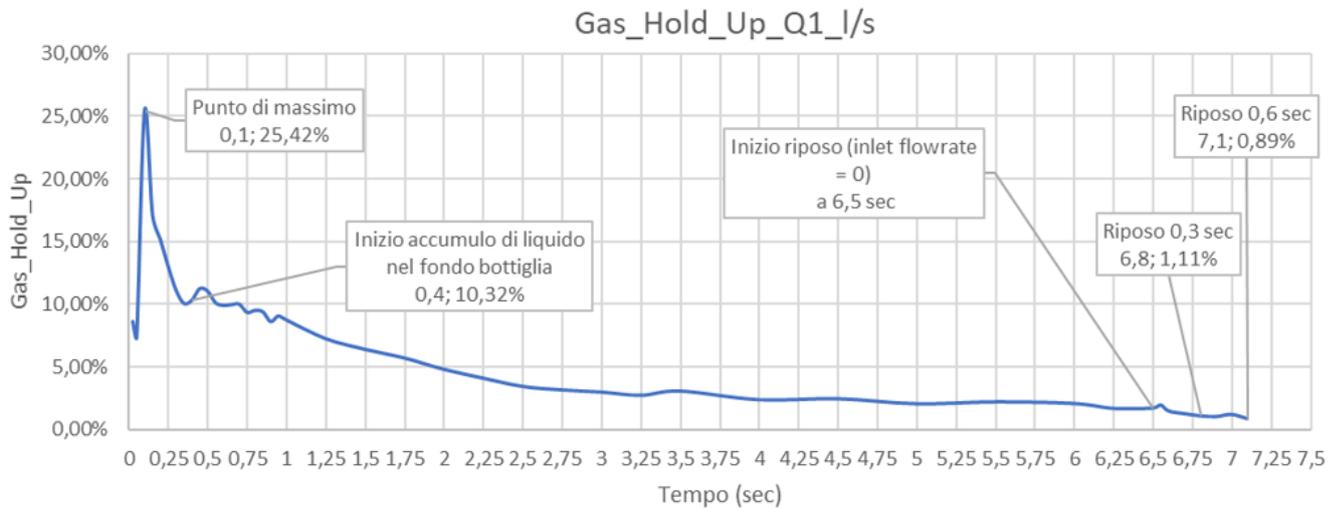
Dall'analisi dei grafici emerge come il contenuto di gas all'interno del battente liquido diminuisca nel tempo. Inoltre, la perdita di gas accelera quando viene fermato l'ingresso di liquido, ovvero quando inizia la fase di riposo.



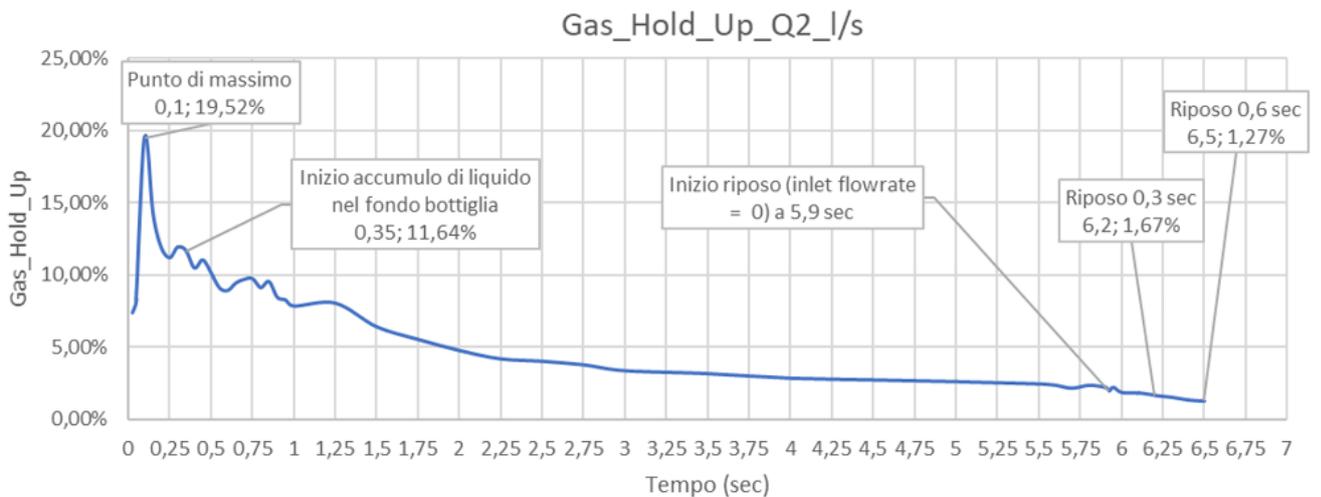
**Figura 6.3:** Portata  $Q_1$ , `alpha.water`, time step 1 secondo.  
Sinistra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo pre-Threshold.  
Destra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo post-Threshold.



**Figura 6.4:** Portata  $Q_1$ , `alpha.water`, time step 6 secondi.  
Sinistra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo pre-Threshold.  
Destra: la vista in sezione interna alla bottiglia mostra il dominio di calcolo post-Threshold.

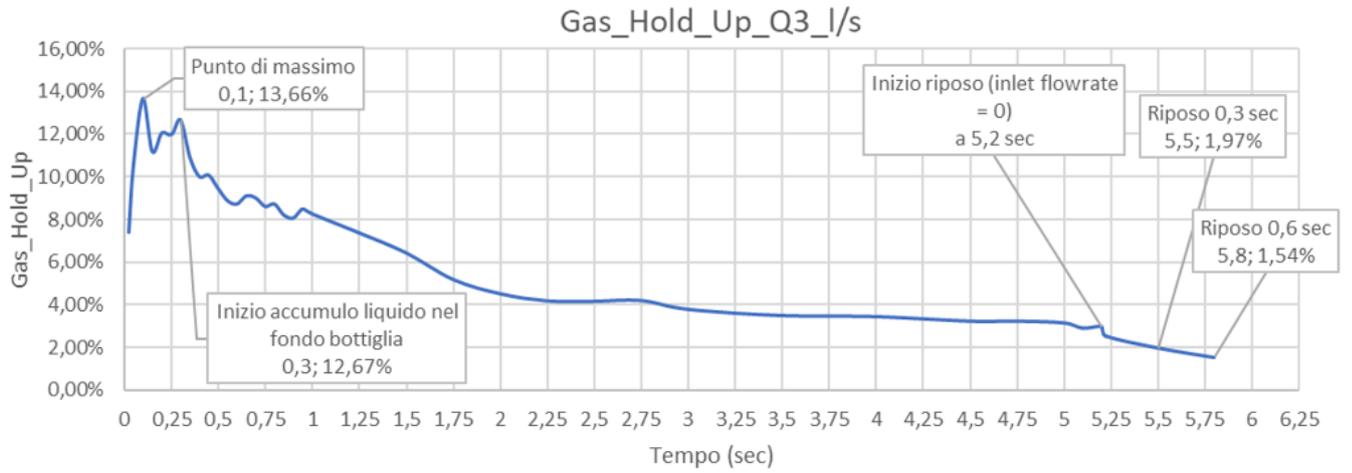


**Figura 6.5:** Portata  $Q_1$ : andamento della frazione di gas all'interno del battente liquido.

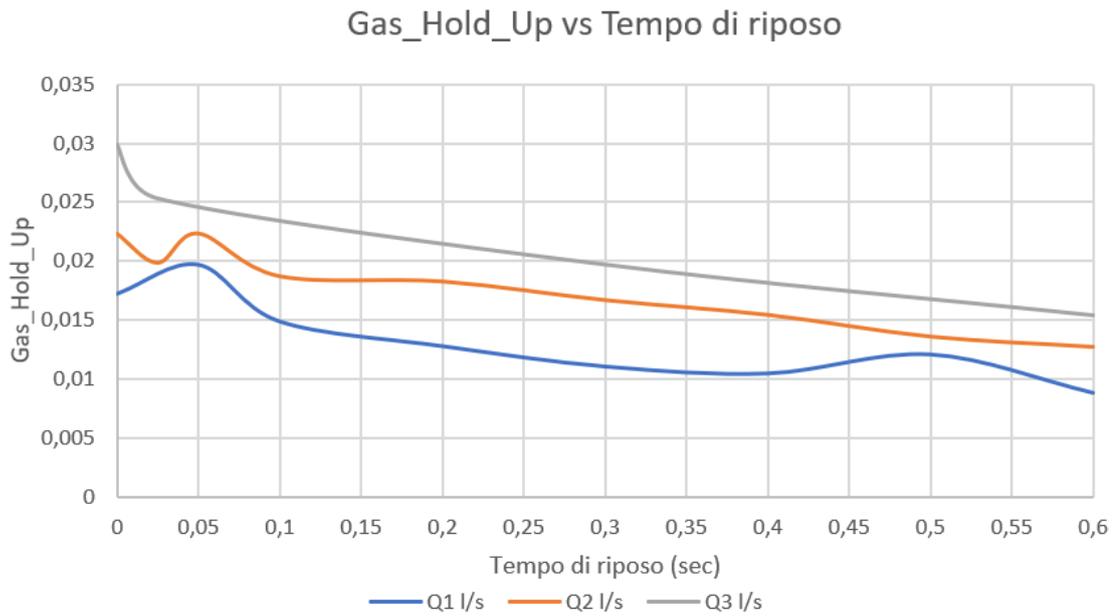


**Figura 6.6:** Portata  $Q_2$ : andamento della frazione di gas all'interno del battente liquido.

Nel grafico in Figura 6.8 viene riportato l'andamento della frazione gassosa intrappolata nel battente liquido per le tre portate all'interno della fase di riposo. Da ciò è possibile osservare come portate maggiori intrappolino più gas, confermando l'intuizione fisica.

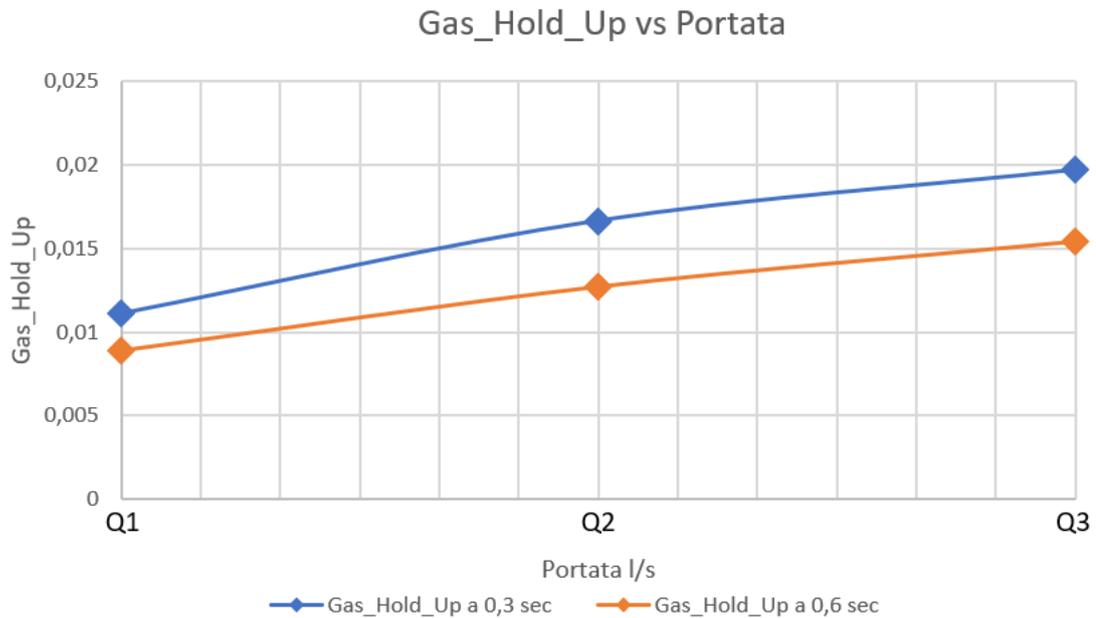


**Figura 6.7:** Portata  $Q_3$ : andamento della frazione di gas all'interno del battente liquido.



**Figura 6.8:** Andamento della frazione gassosa intrappolata nel battente liquido per le tre portate all'interno della fase di riposo.

In aggiunta, all'interno della Figura 6.9, vengono interpolati i risultati riguardanti la frazione di gas intrappolata nel battente liquido per le tre portate, a due diversi



**Figura 6.9:** Frazione di gas intrappolata nel battente liquido prevista interpolando i valori ottenuti dalle simulazioni delle portate  $Q_1$ ,  $Q_2$  e  $Q_3$ .

istanti della fase di riposo. Questo viene eseguito perché nel range di portate investigate sembra esserci una relazione lineare fra portata ed hold-up, come mostrato in Figura 6.9.

### 6.3 Analisi riguardante l'influenza di concentrazione, temperatura e portata di riempimento sulla produzione di schiuma durante la prima fase di degasamento

L'ultima parte del lavoro di tesi ha avuto come obiettivo quello di individuare l'influenza dei vari parametri sulla produzione di schiuma durante la prima fase di

degasamento al seguito del riempimento della bibita gassata.

### 6.3.1 Procedura di degasamento

Come già accennato in precedenza all'interno del Capitolo 4, l'operazione di riempimento di una bibita gassata in bottiglia è costituita da una fase attiva e da una passiva, chiamata riposo. A seguito della fase di riposo avviene il degasamento, il quale è diviso in due parti.

La prima parte prevede una depressurizzazione della bottiglia, la quale viene portata alla pressione atmosferica. Durante questa prima parte si ha una formazione di schiuma dovuta alle bolle di gas intrappolate nel liquido a causa del riempimento. Infatti, la depressurizzazione causa una rapida risalita delle bolle di gas intrappolate all'interno della fase liquida, le quali si espandono provocando una colonna di schiuma. In questa fase è importante che la schiuma che si forma non fuoriesca dalla bottiglia, causando sporco della superficie esterna della bottiglia e producendo una perdita di liquido lungo la linea produttiva.

La seconda parte del degasamento prevede anch'essa una depressurizzazione e anche in questa fase si ha la produzione di una colonna di schiuma, la quale ha, però, altezza minore rispetto alla schiuma formatasi durante la prima fase. In questa seconda fase la produzione della schiuma è dovuta alla nucleazione di nuove bolle e non è stata presa in considerazione.

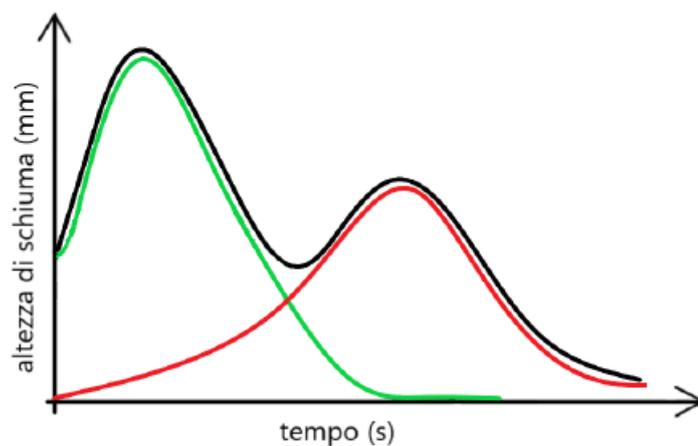
In questa tesi, dunque, l'attenzione è stata posta solo alla schiuma formatasi durante la prima fase di degasamento.

L'intera fase di degasamento è schematizzata all'interno della Figura 6.10.

### 6.3.2 Campagna sperimentale

Sperimentalmente sono state condotte diverse prove riguardanti l'operazione di riempimento della bibita gassata in bottiglia e successivo degasamento. In queste prove sono stati fatti variare tre fattori (temperatura, concentrazione di  $CO_2$  disciolta nel liquido e battente, ovvero la portata di liquido) ed è stata misurata l'altezza della colonna di schiuma formatasi durante la prima fase di degasamento. Per quanto riguarda i fattori, sono state prese in considerazione due temperature ( $T_1$  e  $T_2$ ), tre concentrazioni ( $C_1$ ,  $C_2$  e  $C_3$ ) e tre portate di riempimento ( $Q_1$ ,  $Q_2$  e  $Q_3$ ).

All'interno della Tabella 6.1 è possibile visualizzare le prove sperimentali effettuate.



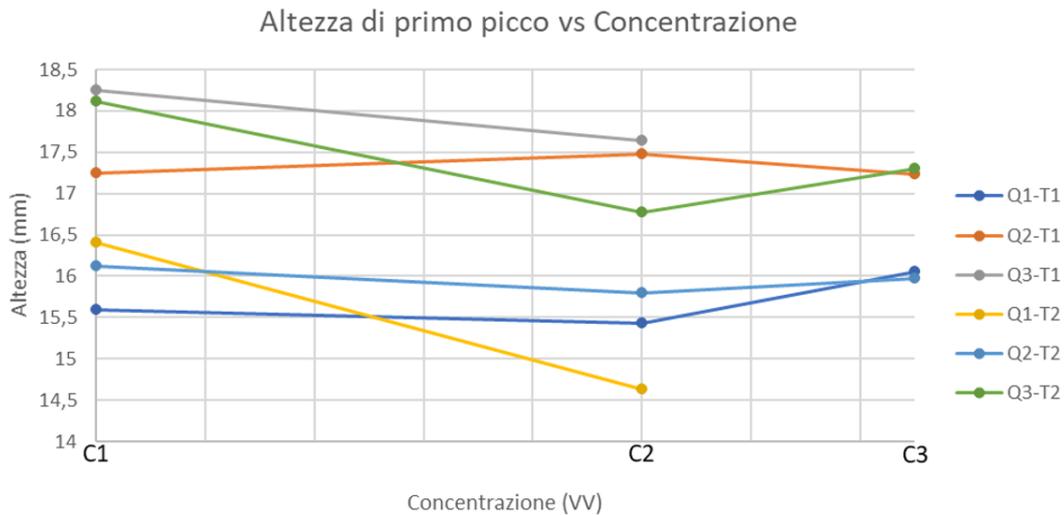
**Figura 6.10:** La curva nera è una rappresentazione approssimata del profilo di schiuma, il quale può essere scomposto idealmente nella curva verde, rappresentante il contributo del riempimento, e nella curva rossa, rappresentante il contributo della seconda fase di degasamento.

| Concentrazione | Portata | Temperatura | Altezza Schiuma (mm) |
|----------------|---------|-------------|----------------------|
| $C_1$          | $Q_1$   | $T_1$       | 15,59                |
| $C_1$          | $Q_1$   | $T_2$       | 16,41                |
| $C_1$          | $Q_2$   | $T_1$       | 17,24                |
| $C_1$          | $Q_2$   | $T_2$       | 16,12                |
| $C_1$          | $Q_3$   | $T_1$       | 18,25                |
| $C_1$          | $Q_3$   | $T_2$       | 18,11                |
| $C_2$          | $Q_1$   | $T_1$       | 15,42                |
| $C_2$          | $Q_1$   | $T_2$       | 14,62                |
| $C_2$          | $Q_2$   | $T_1$       | 17,47                |
| $C_2$          | $Q_2$   | $T_2$       | 15,79                |
| $C_2$          | $Q_3$   | $T_1$       | 17,63                |
| $C_2$          | $Q_3$   | $T_2$       | 16,77                |
| $C_3$          | $Q_1$   | $T_1$       | 16,04                |
| $C_3$          | $Q_2$   | $T_1$       | 17,22                |
| $C_3$          | $Q_2$   | $T_2$       | 15,96                |
| $C_3$          | $Q_3$   | $T_2$       | 17,30                |

**Tabella 6.1:** Collezione di dati sperimentali sul primo picco di schiuma.

### 6.3.3 Influenza dei fattori sul primo picco di schiuma: risultati

Al fine di comprendere l'influenza di temperatura, portata e concentrazione sul primo picco di schiuma, le varie prove sperimentali sono state analizzate mediante una procedura grafica, come è possibile osservare nelle Figure 6.11, 6.12 e 6.13.



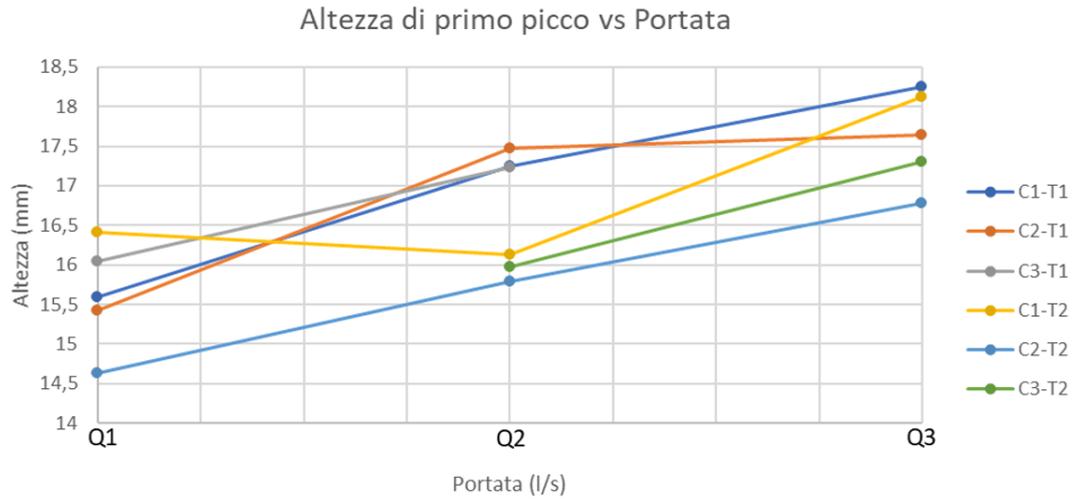
**Figura 6.11:** Altezza di schiuma ottenuta nella prima di fase di degasamento variando la concentrazione e mantenendo costanti la portata e la temperatura.

Dall'analisi di questi tre grafici emerge come la portata sia il fattore che influisce di più sull'altezza di schiuma ottenuta nella prima fase di degasamento. Per quanto riguarda temperatura e concentrazione, il loro effetto è minore e tal volta trascurabile.

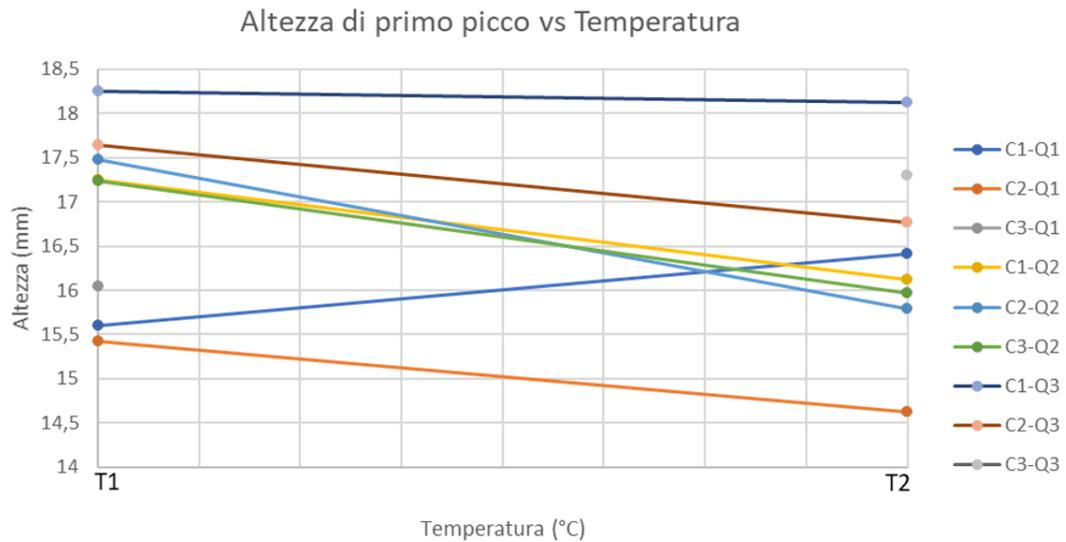
#### Utilizzo del software MODDE

Al fine di realizzare un'analisi più approfondita sui i fattori che influiscono sulla produzione del primo picco di schiuma, è stato utilizzato il software MODDE, sviluppato da Umetrics. Tramite questo software, utilizzando i dati sperimentali, si è ricavato il modello matematico che lega fattori e risposte. Questo modello matematico, poi, è stato analizzato mediante due strumenti: **Coefficients Plot** e **Vip plot**.

Lo strumento **Coefficients Plot** permette di correlare la variazione della risposta, ovvero l'altezza di schiuma, alla variazione di uno dei tre fattori in esame. In Figura 6.14 è riportata l'interfaccia grafica prodotta dall'esecuzione dello strumento **Coefficients Plot** nel caso dei dati sperimentali. All'interno dell'istogramma l'altezza della barra, ovvero il coefficiente, indica la variazione nella risposta quando



**Figura 6.12:** Altezza di schiuma ottenuta nella prima di fase di degasamento variando la portata e mantenendo costanti la concentrazione e la temperatura.



**Figura 6.13:** Altezza di schiuma ottenuta nella prima di fase di degasamento variando la temperatura e mantenendo costanti la concentrazione e la portata.

il fattore associato al coefficiente varia di un'unità, mentre gli altri fattori sono mantenuti ad un valore costante medio. Dunque, il modello indica una maggiore sensibilità nei confronti della portata piuttosto che nei confronti di concentrazione e temperatura. MODDE indica, inoltre, che l'altezza è direttamente proporzionale

alla portata e inversamente proporzionale a concentrazione e temperatura. Nei confronti di temperatura e concentrazione la variazione è così piccola che è possibile trascurare il loro effetto.

Per quanto riguarda lo strumento *Vip plot*, esso ha l'unica funzione di sintetizzare

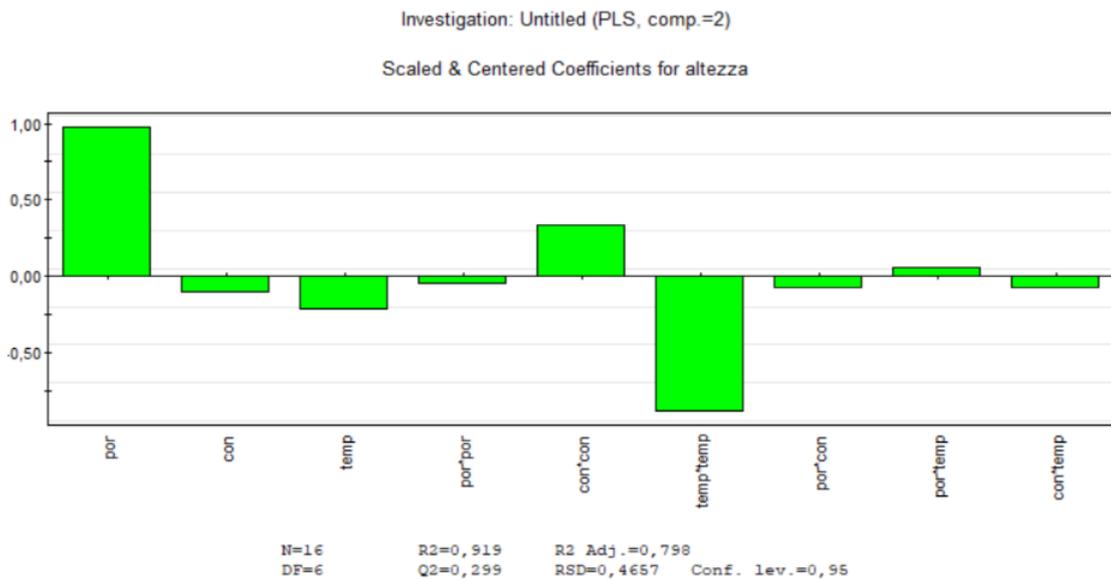


Figura 6.14: MODDE: Coefficients Plot.

l'importanza dei vari termini all'interno del modello. Anche in questo caso viene riportata (Figura 6.15) l'interfaccia grafica prodotta dall'esecuzione dello strumento *Vip plot* nel caso dei dati sperimentali. Dall'istogramma prodotto emerge come la portata sia il termine più importante per quanto riguarda l'altezza di schiuma nella prima fase di degasamento.

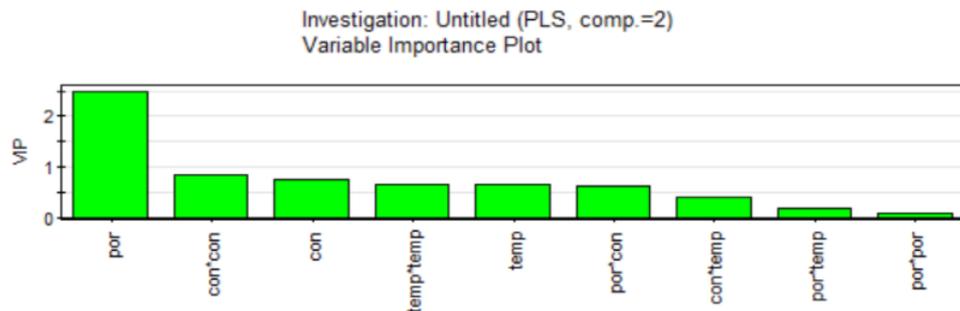


Figura 6.15: MODDE: Vip plot.

| Portata | Altezza di schiuma calcolata (mm) |
|---------|-----------------------------------|
| $Q_1$   | 4,6                               |
| $Q_2$   | 6,9                               |
| $Q_3$   | 8,4                               |

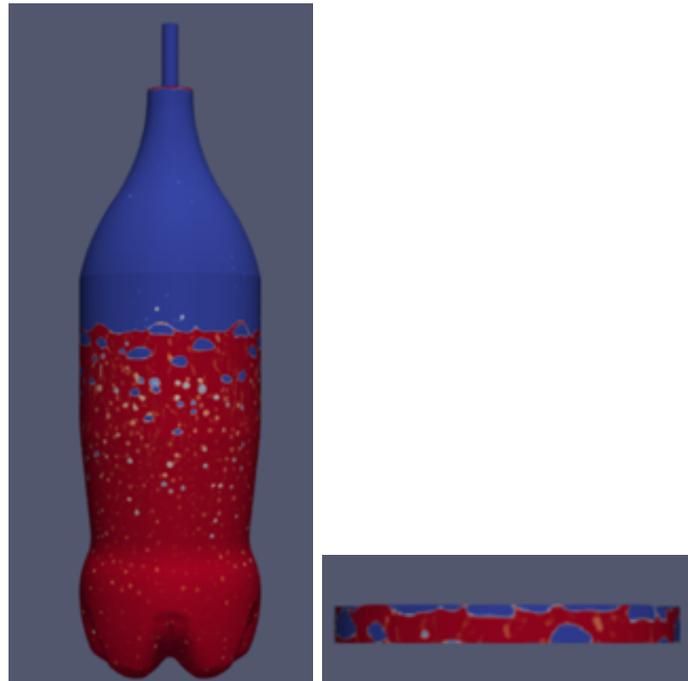
**Tabella 6.2:** Altezza di primo picco calcolata sulla base delle simulazioni effettuate.

## 6.4 Tentativo di ottenimento dell'altezza di schiuma nella prima fase di degasamento tramite espansione del gas intrappolato

In conclusione al lavoro di tesi, è stato fatto un tentativo di calcolo dell'altezza di schiuma raggiunta durante la prima fase di degasamento per le tre portate simulate, ovvero  $Q_1$ ,  $Q_2$  e  $Q_3$ . In pratica, si è voluta calcolare l'altezza di schiuma dovuta all'espansione del volume di gas, da pressione  $P_{tank}$  ad un'atmosfera, contenuto in un volumetto di altezza 1 cm sotto l'interfaccia gas-liquido a 0,3 secondi di riposo. Per realizzare ciò, tramite **Paraview**, per ciascuna portata è stato considerato il time step corrispondente alla fine del riposo di 0,3 secondi. Poi, è stata attuata la seguente procedura:

- tramite comando `clip` considerare una porzione di volume di 1 cm sotto l'interfaccia. L'ottenimento di questa porzione di volume è osservabile in Figura 6.16;
- su questa porzione di volume, tramite `integrateVariables`, si calcola il volume e la frazione di gas contenuta in essa. Si ricava dunque il volume di gas all'interno delle regione di un centimetro sotto l'interfaccia;
- il volume di gas ottenuto al passo precedente lo si fa espandere da  $P_{tank}$  ad un'atmosfera tramite la legge dei gas perfetti;
- conoscendo la sezione della porzione di volume, si calcola l'altezza di espansione dividendo il volume di gas dopo l'espansione per la sezione della fettina.

In Tabella 6.2 è possibile osservare l'altezza di schiuma calcolata per le tre portate simulate. Si noti come all'aumentare della portata simulata l'altezza di schiuma aumenti e ciò è in linea con le considerazioni fatte precedentemente.



**Figura 6.16:** Sinistra: time step corrispondente alla fine del riposo di 0.3 secondi per una delle tre portate.  
Destra: l'applicazione del comando `clip` permette l'ottenimento di una porzione di volume di spessore 1 cm situata sotto l'interfaccia.

# Capitolo 7

## Conclusioni

Tenendo conto del lavoro svolto in questa tesi, è possibile esprimere alcune considerazioni riguardo all'impiego della fluidodinamica computazionale per la simulazione del processo di riempimento di bevande gassate in bottiglia.

In primo luogo, è possibile evidenziare il successo della metodologia per simulare la valvola rotativa e per ricavarne da essa le condizioni al contorno per la simulazione successiva di riempimento in bottiglia della bibita gassata. Questa procedura ha, infatti, permesso di ottenere una simulazione del tutto compatibile con ciò che si è osservato sperimentalmente.

La principale difficoltà incontrata nel lavoro di questa tesi è stata la descrizione del film liquido aderente alla parete della bottiglia. Infatti, al fine di avere un'accurata risoluzione di un sottile strato di liquido in movimento e a contatto con una parete è necessario l'utilizzo di molte celle computazionali e ciò rende l'operazione di simulazione parecchio onerosa sia in termini di tempo che in termini di risorse informatiche. Questa è una limitazione del metodo VOF, che potrebbe essere risolta usando altri metodi per il tracciamento dell'interfaccia.

Inoltre, il lavoro svolto evidenzia come l'utilizzo della fluidodinamica computazionale abbia molte potenzialità all'interno dell'industria alimentare. Infatti, essa permette di ottenere diverse informazioni riguardo ad alcuni aspetti del processo industriale difficilmente ricavabili con precisione dal punto di vista empirico, come ad esempio lo spessore del film liquido aderente alla parete della bottiglia e la frazione di gas intrappolato all'interno del liquido al fondo della bottiglia in ogni istante temporale del processo di riempimento di una bibita gassata. Inoltre, è possibile ottenere anche la dimensione delle bolle di gas intrappolate all'interno del liquido, così come la forma dell'interfaccia gas-liquido in ogni istante temporale e in ogni zona interna alla bottiglia.

Per quanto riguarda l'ultima parte del lavoro svolto, ovvero il tentativo di ottenere dalle simulazioni realizzate l'altezza di schiuma raggiunta nella prima fase dell'operazione di degasamento della bibita, è possibile constatare i limiti della metodologia

Volume of Fluid (VOF). Inoltre, è possibile affermare come l'impiego di metodi di tracciamento dell'interfaccia gas-liquido più complessi e precisi del Volume of Fluid Method (VOF), come ad esempio il metodo level-set, o l'adozione di una griglia computazionale mobile possano ulteriormente migliorare la simulazione del processo di riempimento di una bibita gassata in bottiglia e, dunque, migliorare la predizione del livello di schiuma raggiunto durante le prime fasi di degasamento successive al riempimento.

# Bibliografia

- [1] Cyril W Hirt e Billy D Nichols. «Volume of fluid (vof) method for the dynamics of free boundaries». In: *Journal of computational physics* 39.1 (1981), pp. 201–225 (cit. alle pp. 19–21).
- [2] Onno Ubbink. «Numerical prediction of two fluid systems with sharp interfaces». Tesi di dott. University of London, 1997 (cit. a p. 21).
- [3] H.Weller. «A new approach to vof-based interface capturing methods for incompressible and compressible flow». In: *Technical Report TR/HGW/04* (2008) (cit. a p. 25).
- [4] G.Liger-Belair. «Derivation, modelling and solution of the conditionally averaged two-phase flow equations». In: *Nabla Ltd, No Technical Report TR/HGW 2* (2002).
- [5] Henrik Rusche. «Computational fluid dynamics of dispersed two-phase flows at high phase fractions». Tesi di dott. Imperial College London(University of London), 2003.
- [6] C. J. Greenshields. «OpenFOAM user guide». In: *OpenFOAM Foundation Ltd, version 8* (2020).
- [7] *Introduzione a OpenFOAM: introduzione a software e tutorial gratuiti*. <https://conoscerelinux.org/wp-content/uploads/2016/06/openfoam-introduzione-corso-conoscerelinux-modena-2016.pdf>. Visited: 2020-11-08.
- [8] V.H. Bhusare, M.K. Dhiman, D.V. Kalaga, S. Roy e J.B. Joshi. «CFD simulations of a bubble column with and without internals by using OpenFOAM». In: *Chemical Engineering Journal* 317 (2017), pp. 157–174.
- [9] Pedro Lopes. «Free-surface flow interface and air-entrainment modelling using OpenFOAM». Tesi di dott. University of Coimbra PhD Thesis, 2003.
- [10] D. Steen e P. R. Ashurst. *Carbonated soft drinks: formulation and manufacture*. John Wiley & Sons, 2008.
- [11] *Coca Cola-Storia*. <https://www.coca-colaitalia.it/il-nostro-mondo/storia>. Visited: 2020-11-02.

- 
- [12] Hrvoje Jasak. «Error analysis and estimation for finite volume method with applications to fluid flow.» Tesi di dott. Imperial College London (University of London), 1996.
- [13] A.D. Ronteltap. «Beer foam physics». Tesi di dott. Wageningen University & Research, 1989.
- [14] E. Salerno. «Fluidodinamica in bottiglia: la formazione di schiuma nel processo di imbottigliamento di bevande gassate». Tesi di dott. University of Modena e Reggio Emilia, 2015.
- [15] S.G. Bankoff. «Entrapment of gas in the spreading of a liquid over a rough surface». In: *AIChE journal* 4.1 (1958), pp. 24–26.
- [16] T. J. Lin e H.G. Donnelly. «Gas bubble entrainment by plunging laminar liquid jets». In: *AIChE Journal* 12.3 (1966), pp. 563–571.
- [17] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo e B. Van Wachem. *Computational fluid dynamics for engineers*. Cambridge university press, 2011.
- [18] F. Moukalled, L. Mangani, M. Darwish et al. *The finite volume method in computational fluid dynamics*. Vol. 6. Springer, 2016.
- [19] J.H. Ferziger, M. Perić e R. L. Street. *Computational methods for fluid dynamics*. Vol. 3. Springer, 2002.
- [20] H.K. Versteeg e W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.
- [21] Y. Liu. «Two-Fluid Modeling of Gas-Solid and Gas-Liquid Flows: Solver Development and Application». Tesi di dott. Technische Universität, 2014.
- [22] H.Weller. «Controlling the computational modes of the arbitrarily structured C grid». In: *Monthly Weather Review* 140.1 (2012), pp. 3220–3234.
- [23] H. Tang e L.C. Wrobel. «Modelling the interfacial flow of two immiscible liquids in mixing process». In: *International Journal of Engineering Science* 43.15 (2005), pp. 1234–1256.