UNIVERSITÀ DEGLI STUDI DI GENOVA

SCUOLA POLITECNICA

DIME

Dipartimento di Ingegneria Meccanica, Energetica, Gestionale e dei Trasporti



TESI DI LAUREA MAGISTRALE
IN
INGEGNERIA MECCANICA - ENERGIA E AERONAUTICA

Analisi della sensitività e ottimizzazione dell'ala anteriore di una vettura da competizione basata sul metodo dell'aggiunto utilizzando OpenFOAM

Relatore:
Chiar.mo Prof. Ing. Jan Oscar Pralits
Chiar.mo Ing. Edoardo Alinovi

Allievo:
Francesco Girardi

Luglio 2025

Abstract

The following discussion primarily aims to make an adjoint-based optimisation of the front wing of an open wheel formula car. We examined the front wing and wheel, as the wheel's effect on the wing is significant, in a simplified configuration using two different meshes. We compared the results obtained with OpenFOAM to those from Ansys Fluent. After, a comparison was made between the results from the $k - \omega$ SST and Spalart-Allmaras turbulence models, then we proceeded to the calculation of sensitivity maps and the final shape optimization of the geometry. The objective of the deformation, based on the continuous adjoint implemented in OpenFOAM and the Free Form Deformation technique, was to maximize the downforce generated by the wing. Finally, the results obtained before and after the geometry's shape variation were studied and compared. Key aspects of this thesis include the collaboration with the racing car manufacturer Dallara Automobili S.p.A., which provided the initial geometry and mesh, and the use of the open-source computational fluid dynamics software OpenFOAM, for preliminary calculations and the final geometry optimization.

Sommario

La seguente trattazione si pone come obiettivo principale l'ottimizzazione basata sul metodo dell'aggiunto dell'ala anteriore di una vettura a ruote scoperte tipo formula. Sono state esaminate ala anteriore e ruota, essendo l'effetto di questa sull'ala non trascurabile, in una configurazione semplificata, su due mesh distinte. Si sono confrontati i risultati ottenuti con OpenFOAM con quelli ricavati da Ansys Fluent, successivamente è stato fatto un confronto tra i risultati ottenuti con i modelli di turbolenza $k-\omega$ SST e Spalart-Allmaras, per poi passare al calcolo delle mappe di sensitività e all'ottimizzazione di forma della geometria. L'obiettivo della deformazione, eseguita utilizzando l'aggiunto continuo implementato in OpenFOAM e la tecnica Free Form Deformation, è stato la massimizzazione della deportanza generata dall'ala. Si sono infine studiati e confrontati i risultati ottenuti prima e dopo la variazione di forma della geometria. Peculiarità del lavoro di tesi sono sicuramente la collaborazione con l'azienda costruttrice di auto da corsa Dallara Automobili S.p.A., che ha fornito geometria e mesh di partenza, e l'utilizzo del programma di calcolo open source OpenFOAM per i calcoli preliminari e l'ottimizzazione finale della geometria.

Ringraziamenti

Desidero ringraziare tutti i miei familiari, in particolare i miei genitori e i miei nonni, per avermi permesso di compiere questo meraviglioso percorso di studi e per il loro costante supporto, essenziale per il raggiungimento di questo traguardo.
Ringrazio Benedetta, per essere stata in particolare negli ultimi due anni un faro nei miei numerosi momenti di difficoltà, per avermi spronato a dare sempre il meglio di me e per essermi sempre accanto.
Ringrazio Matte, Manzo, Big, Dario, Jack, Lollo, Vitto, Pie, Peppo, Anna, Cri e Annalisa, colleghi incredibilmente professionali e soprattutto amici con cui ho condiviso gli studi e tanti momenti che mi rimarranno sempre nel cuore.
Ringrazio Ilaria e Francesco, per tutte le belle giornate e i momenti passate insieme.
Ringrazio Simo, Nicco, Edo, Sam, Luca e Alfo per tutte le avventure passate insieme e per quelle future che sono sicuro arriveranno.
Ringrazio il TooBar di Varazze per il supporto fondamentale e per essere stato un punto di riferimento per molte serate in compagnia dei miei amici.
Ringrazio tutti gli amici di Morbello, che mi sono vicini da sempre e con i quali ho condiviso bellissimi momenti che porterò sempre con me.
Ultimi, ma non per importanza, ringrazio Jan, Edoardo, Elisa e Mattia per l'incredibile opportunità di collaborazione che mi è stata concessa, per tutto il supporto ricevuto durante lo svolgimento del lavoro di tesi, per il bagaglio di esperienza che mi rimarrà sempre e per la splendida giornata passata insieme presso la sede di Dallara Automobili S.p.A. che porterò sempre con me.



Indice

1 Introduzione **1**
- 1.1 Motivazioni e obiettivi . . . 1
- 1.2 Sviluppo dell'aerodinamica nelle macchine da corsa . . . 1
  - 1.2.1 Portanza di un profilo e downforce . . . 3
  - 1.2.2 Aerodinamica odierna delle auto da corsa . . . 4
- 1.3 Dallara Automobili S.p.A. . . . 5

2 Fluidodinamica Computazionale (CFD) e OpenFOAM **7**
- 2.1 Computational Fluid Dynamics (CFD) . . . 7
- 2.2 OpenFOAM . . . 8
  - 2.2.1 Cenni storici . . . 8
- 2.3 DAFoam . . . 9
- 2.4 Struttura di OpenFOAM . . . 10
  - 2.4.1 Directory "0" . . . 10
  - 2.4.2 Directory "constant" . . . 11
  - 2.4.3 Directory "system" . . . 11

3 Equazioni governanti **12**
- 3.1 Equazioni di Navier-Stokes . . . 12
- 3.2 Modelli di turbolenza . . . 13
  - 3.2.1 Modelli RANS . . . 14

4 Geometria, dominio di calcolo, "mesh" e condizioni al contorno **20**
- 4.1 Geometria, "patch" dell'auto e dominio di calcolo . . . 20
- 4.2 Griglie di calcolo (mesh) . . . 23
  - 4.2.1 Determinazione del "first layer thickness" . . . 25
- 4.3 Condizioni al contorno . . . 26
  - 4.3.1 Directory "0" . . . 26
  - 4.3.2 Directory "constant" . . . 35
  - 4.3.3 Directory "system" . . . 37

5 Risultati del calcolo diretto **42**
- 5.1 Risultati iniziali e modifiche alle griglie di calcolo . . . 42
- 5.2 Confronto tra OpenFOAM e Ansys Fluent . . . 47
  - 5.2.1 Confronto dei risultati per la griglia di calcolo "hyp" . . . 48
  - 5.2.2 Confronto dei risultati per la griglia di calcolo "lyp" . . . 52
- 5.3 Risultati Spalart-Allmaras . . . 56

5.3.1 Distribuzione di "$\mathbf{y}+$" e del Coefficiente di Pressione (C_p) 56

6 Metodo dell'aggiunto **60**
6.1 Analisi della sensitività e metodo dell'aggiunto 60
6.1.1 L'operatore aggiunto 62
6.2 Equazioni aggiunte applicate al caso di Navier-Stokes 64
6.3 Metodo dell'aggiunto: setup di OpenFOAM 66
6.3.1 Directory "0" 66
6.3.2 Directory "constant" 69
6.3.3 Directory "system" 69

7 Ottimizzazione **75**
7.1 Primi risultati 75
7.1.1 Risoluzione dei problemi 77
7.1.2 Utilizzo di Spalart-Allmaras per il calcolo dell'aggiunto 78
7.2 Mappe di sensitività 80
7.2.1 Mappe di sensitività, $J = C_l$ 81
7.2.2 Mappe di sensitività, $J = C_l + 5C_d$ 82
7.3 Processo di ottimizzazione e deformazione della geometria 86
7.4 Ottimizzazione di forma finale 88
7.4.1 Confronto tra le distribuzioni di C_p prima e dopo la deformazione 90
7.4.2 Confronto contour di Coefficiente di Pressione e Coefficiente di Pressione Totale 98

8 Conclusioni **105**

Elenco delle figure

1.1 Jacky Ickx Gran Premio di Spagna 1969 - Brabham BT26A 2
1.2 Esempio di profilo alare e forze agenti su di esso 3
1.3 Jacques Laffite Gran Premio d'Austria 1982 - Ligier JS19-3 4
1.4 Ala anteriore Ferrari 312 B3 1974 4
1.5 Ala anteriore Haas VF-18 2018 5
1.6 I° Trofeo Cinquetti 1973 - Dallara SP 1000 6

4.1 Geometria iniziale divisa in singole patch 21
4.2 Dominio di calcolo diviso in singole patch 22
4.3 Dettaglio dei corpi di influenza e relativi infittimenti 23
4.4 Boundary layer mesh hyp 24
4.5 Boundary layer mesh lyp 24
4.6 Contour y+ mesh hyp 24
4.7 Contour y+ mesh lyp 24

5.1 Soluzione "Gauss linearUpwind"
Coefficiente di "lift": $C_l = -0.85$ 43
5.2 Soluzione "Gauss upwind"
Coefficiente di "lift": $C_l = -0.55$ 43
5.3 Ruota nella posizione originale 45
5.4 Ruota nella posizione modificata 45
5.5 Sezioni di controllo sull'ala 47
5.6 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.37$ 49
5.7 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.42$ 49
5.8 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.53$ 50
5.9 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.63$ 50
5.10 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.74$ 51
5.11 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.79$ 51
5.12 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.37$ 53
5.13 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.42$ 53
5.14 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.53$ 54
5.15 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.63$ 54
5.16 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.74$ 55
5.17 Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.79$ 55
5.18 Contour y+ $k - \omega$ SST 56
5.19 Contour y+ Spalart-Allmaras 56
5.20 Confronto del C_p tra modello $k - \omega$ SST e Spalart-Allmaras - $y/s = 0.37$ 57
5.21 Confronto del C_p tra modello $k - \omega$ SST e Spalart-Allmaras - $y/s = 0.42$ 57

5.22 Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.53$ 58
5.23 Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.63$ 58
5.24 Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.74$ 59
5.25 Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.79$ 59

7.1 Residui calcolo diretto $k-\omega$ 76
7.2 Continuità calcolo diretto $k-\omega$ 76
7.3 Residui calcolo aggiunto $k-\omega$ 76
7.4 Continuità calcolo aggiunto $k-\omega$ 76
7.5 Residui calcolo aggiunto adjointLaminar 78
7.6 Continuità calcolo aggiunto adjointLaminar 78
7.7 Residui calcolo diretto Spalart-Allmaras 78
7.8 Continuità calcolo diretto Spalart-Allmaras 78
7.9 Residui calcolo aggiunto adjointSpalartAllmaras 79
7.10 Continuità calcolo diretto adjointSpalartAllmaras 79
7.11 Mappa di sensitività $J = C_l$ - adjointLaminar 81
7.12 Mappa di sensitività $J = C_l$ - adjointSpalartAllmaras 81
7.13 Mappa di sensitività - $J = C_l + 5C_d$ - adjointLaminar 82
7.14 Mappa di sensitività - $J = C_l + 5C_d$ - adjointSpalartAllmaras 82
7.15 Sezione dell'ala utilizzata per l'analisi della sensitività 83
7.16 Sensitività e direzione di deformazione - $J = C_l$ - adjointLaminar 84
7.17 Sensitività e direzione di deformazione - $J = C_l$ - adjointSpalartAllmaras 84
7.18 Sensitività e direzione di deformazione - $J = C_l + 5C_d$ - adjointLaminar 85
7.19 Sensitività e direzione di deformazione - $J = C_l + 5C_d$ - adjointSpalartAllmaras . . 85
7.20 Box di deformazione visuale frontale 86
7.21 Box di deformazione visuale dall'alto 86
7.22 Deformazione iniziale - $J = C_l$ - adjointSpalartAllmaras, adjointLaminar 87
7.23 Deformazione iniziale - $J = C_l + 5C_d$ - adjointSpalartAllmaras, adjointLaminar . . 87
7.24 C_l vs Optimisation step - "adjointLaminar" 89
7.25 C_l vs Optimisation step - "AdjointSpalartAllmaras" 89
7.26 Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.37$ 91
7.27 Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.42$ 92
7.28 Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.53$ 93
7.29 Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.63$ 94
7.30 Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.74$ 95
7.31 Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.79$ 96
7.32 Contour di C_p - Geometria originale vs Geometria modificata - $y/s = 0.42$ 99
7.33 Contour di C_p - Geometria originale vs Geometria modificata - $y/s = 0.63$ 100

7.34 Contour di C_p - Geometria originale vs Geometria modificata - $y/s = 0.79$ 101
7.35 Contour di C_{pT} - Geometria originale vs Geometria modificata - $y/s = 0.42$ 102
7.36 Contour di C_{pT} - Geometria originale vs Geometria modificata - $y/s = 0.63$ 103
7.37 Contour di C_{pT} - Geometria originale vs Geometria modificata - $y/s = 0.79$ 104

8.1 Vista frontale geometria originale 106
8.2 Vista frontale geometria modificata 106

Elenco delle tabelle

3.1 Costanti del modello $k - \omega$ SST 17
3.2 Coefficienti del modello di turbolenza Spalart-Allmaras 19

4.1 Nomenclatura delle patch della geometria iniziale 21
4.2 Nomenclatura delle patch del dominio 22

6.1 Condizioni al contorno della velocità di diretto e aggiunto 67
6.2 Condizioni al contorno della pressione di diretto e aggiunto 67
6.3 Condizioni al contorno dell'energia cinetica turbolenta di diretto e aggiunto 68
6.4 Condizioni al contorno del tasso di dissipazione specifico di diretto e aggiunto 68
6.5 Condizioni al contorno della viscosità turbolenta modificata di diretto e aggiunto . . 68

Nomenclature

$\alpha_{k1}, \alpha_{k2}, \alpha_{\omega 1}, \alpha_{\omega 2}, \beta_1, \beta_2, \gamma_1, \gamma_2, \beta^*, a_1, b_1, c_1$ Coefficienti del modello di turbolenza $k-\omega$ SST

Δt Variazione di tempo

Δx Variazione di posizione

$\mathbf{b}_{ij}$ Tensore di Boussinesq

$\mathbf{I}$ Matrice identità

μ Viscosità dinamica

μ_T Viscosità dinamica turbolenta

∇ Nabla

ν Viscosità cinematica

ν_T Viscosità cinematica turbolenta

ω Tasso di dissipazione specifico della turbolenza

$\overline{S}_{ij}$ Tensore della velocità di deformazione mediato

ρ Densità

τ Attrito viscoso (*Viscous stress*)

τ_R Tensore degli sforzi di Reynolds

τ_w Attrito viscoso a parete

$\tilde{\nu}$ Viscosità cinematica turbolenta modificata

$\tilde{P}_k$ *Production limiter* modello $k-\omega$ SST

$\vec{u}$ Velocità

$C_{b1}, \sigma, C_{b2}, k, C_{w1}, C_{w2}, C_{w3}, C_{v1}, C_{t1}, C_{t2}, C_{t3}, C_{t4}$ Coefficienti del modello di turbolenza Spalart-Allmaras

C_{pT} Coefficiente di Pressione Totale

I_t Intensità turbolenta

J Funzione obiettivo

k Energia cinetica turbolenta

l_t	Scala di lunghezza turbolenta
P	Pressione media
p	Pressione
Re_t	Numero di Reynolds turbolento
S	Misura invariante del tasso di deformazione modello $k-\omega$ SST
S_R	Tensore del tasso di variazione (*Strain-rate tensor* mediato con Reynolds)
t	Tempo
U	Velocità mediata
u', v', w'	Fluttuazione di velocità
u, v, w	Componenti della velocità lungo gli assi x,y e z
u_τ	*Shear velocity*
$y+$	Altezza di cella adimensionalizzata
L	Lagrangiana
R	Equazioni di Navier-Stokes
CFL	Numero di Courant (*Courant-Friedrichs-Lewy condition*)
DNS	*Direct Numerical Simulation*
hyp	*High Y Plus*
LES	*Large Eddy Simulation*
lyp	*Low Y Plus*
RANS	*Reynolds Average Navier-Stokes equation*
URF	Fattori di rilassamento (*Under Relaxation Factor*)

Capitolo 1

Introduzione

All'interno di questo capitolo saranno presenti motivazioni e obiettivi del seguente lavoro di tesi, vari cenni storici inerenti ai temi trattati e una serie di capitoli atti ad introdurre lo studio presente nel seguente elaborato, quindi, cenni di aerodinamica e l'utilizzo di ali su vetture da competizione. Sarà inoltre presente qualche breve cenno storico a proposito di Dallara Automobili S.p.A.

1.1 Motivazioni e obiettivi

Il seguente lavoro di tesi è nato in collaborazione con Dallara Automobili S.p.A. con l'obiettivo di studiare una geometria composta da ala e ruota anteriore di una vettura da corsa a ruote scoperte. Lo studio si pone come obiettivo quello di ottenere una mappa di sensitività dell'ala e successivamente ottimizzare la forma di questa utilizzando interamente OpenFOAM, un programma open-source per la fluidodinamica computazionale (CFD). Oltre allo studio e analisi della geometria il lavoro è stato quindi anche utile a testare il comportamento di OpenFOAM quando applicato ad un caso industriale come quello in esame, confrontando i risultati con quelli ricavati da Ansys Fluent, software ampiamente conosciuto e utilizzato in ambito industriale. Inoltre, è servito per fornire a Dallara un metodo validato e funzionante per poter ricavare la mappa di sensitività di una geometria rispetto ad una funzione obiettivo prestabilita ed effettuarne l'ottimizzazione di forma, utilizzando un software privo di licenza quale OpenFOAM.

1.2 Sviluppo dell'aerodinamica nelle macchine da corsa

Storicamente sulle prime auto da corsa prodotte il concetto di aerodinamica della vettura era legato unicamente al cercare di ridurre il più possibile la resistenza all'avanzamento di questa. Successivamente alla seconda guerra mondiale lo sviluppo tecnologico delle auto si è concentrato solo su motore, gomme e telaio; per osservare i primi veri sviluppi aerodinamici di una vettura sarà necessario aspettare fino agli anni '60, quando si iniziano ad osservare vetture aerodinamicamente più raffinate. Questo è avvenuto perché sviluppando i propulsori si sono raggiunte potenze molto elevate e sono emersi i primi problemi legati alla tenuta di strada. Per questo motivo è stato necessario cercare un sistema per aumentare la spinta della macchina verso il terreno in modo da ottenere una maggiore aderenza tra questo e gli pneumatici.

Per percorrere una curva è infatti necessario generare una forza centripeta (verso il centro della curva) per far si che l'auto effettivamente percorra la curva e non proceda in linea retta. Questa forza viene esercitata dall'attrito tra pneumatici e terreno ed è direttamente correlata, tra altre cose quali superficie di contatto e tipo di pneumatico utilizzato, alla forza con cui le ruote vengono schiacciate al suolo.

L'interesse principale è stato quindi quello di cercare un sistema per aumentare la forza con cui la macchina viene spinta verso il terreno senza aumentarne il peso ovviamente, per evitare una perdita di prestazioni nelle altre fasi come quella di frenata o di accelerazione e, più in generale, in tutte quelle situazioni in cui è vantaggioso mantenere un'inerzia del veicolo più bassa possibile. Nasce quindi l'aerodinamica delle autovetture non più con il solo obiettivo di ridurre la resistenza all'avanzamento, ma anche di aumentare la forza con la quale la vettura viene spinta verso il suolo in modo da aumentare l'aderenza degli pneumatici e conseguentemente la velocità di percorrenza delle curve.
Nei intorno agli anni '60 cominciano quindi a comparire le prime ali sulle vetture da competizione inizialmente con forme molto grossolane e posizionate proprio in corrispondenza delle ruote, come visibile in figura 1.1.



Figura 1.1: Jacky Ickx Gran Premio di Spagna 1969 - Brabham BT26A

Oggi l'aerodinamica delle vetture da competizione ha raggiunto livelli estremi ed è in continua evoluzione grazie a tecnologie quali gallerie del vento tecnologicamente molto avanzate e l'introduzione delle simulazioni CFD che permettono di simulare e stimare il comportamento delle vetture in fase di progettazione, senza dover ogni volta costruire costosi prototipi o modelli fisici [1].

1.2.1 Portanza di un profilo e downforce

Quando si parla di ali la prima cosa a cui si pensa sono gli aeromobili, in questo caso le ali vengono utilizzate per produrre una forza verso l'alto che aumenta all'aumentare della velocità dell'aereo. Consideriamo una sezione generica dell'ala di un aereo, visibile in figura 1.2, che si muove ad una certa velocità in un fluido, generalmente aria. La forma del profilo fa si che il flusso abbia una velocità superiore nella parte alta del profilo, generalmente denominata estradosso, e una minore nella parte bassa, generalmente denominata intradosso. Questa differenza di velocità genera a sua volta nel rispetto della Legge di Bernoulli una differenza di pressione tra parte superiore e inferiore, questo genera una spinta verso l'alto (lift o portanza). Sfortunatamente questo non avviene gratuitamente, infatti la generazione di lift da parte del profilo e la viscosità non nulla del fluido generano un'altra forza parallela alla direzione in cui si sta muovendo il corpo, il drag o resistenza all'avanzamento. In ambito automobilistico questo è ovviamente un effetto indesiderato in quanto riduce la velocità dell'auto e conseguentemente la forza verticale generata dal profilo, tuttavia, è possibile andare a studiare la forma della vettura in altre zone, diverse da quelle delle ali, in modo da ridurre il drag complessivo il più possibile. La forza risultante sul profilo sarà quindi la combinazione di lift e drag.

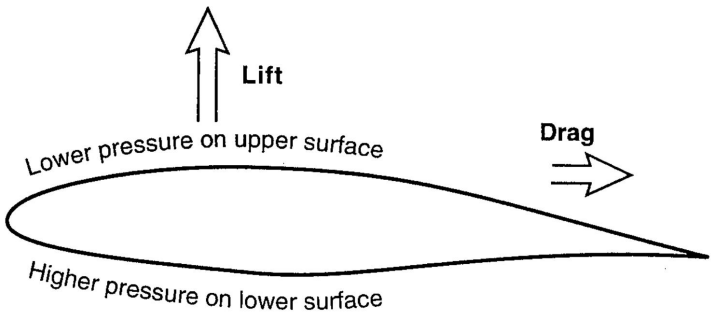


Figura 1.2: Esempio di profilo alare e forze agenti su di esso

Nell'ambito delle corse automobilistiche come detto precedentemente il nostro interesse è quello di spingere la vettura verso il suolo, per fare questo semplicemente si prende il profilo appena descritto e lo si capovolge, in questo modo il lift sarà diretto verso il basso. In gergo automobilistico questa forza prende il nome di "downforce". Il seguente elaborato, in particolare, studia l'ala anteriore di una vettura. Questa, tra le altre cose è caratterizzata dalla sua vicinanza con il terreno e si genera quindi il cosiddetto effetto suolo. Questo fenomeno diventa rilevante nel momento in cui la distanza tra terreno e profilo è inferiore alla corda di quest'ultimo. Avviene in quanto i disturbi di pressione e velocità generati dall'estradosso del profilo non si possono propagare oltre al vincolo imposto dal terreno e questo provoca un ulteriore aumento della velocità sull'estradosso del profilo con conseguente riduzione della pressione e aumento di downforce. Lo studio di questo fenomeno più che sulla forma della ali intorno agli anni '70 ha influito sullo sviluppo dei corpi vettura che venivano progettati in modo da produrre essi stessi il maggior quantitativo di deportanza possibile. Da qui il termine "wing car" nato verso la fine degli anni '70 che stava ad indicare la particolare progettazione del corpo vettura che veniva studiato per essere più simile possibile ad una grossa ala 1.3.



Figura 1.3: Jacques Laffite Gran Premio d'Austria 1982 - Ligier JS19-3

Si noti in questo caso l'assenza dell'ala anteriore volta a lasciare un flusso più pulito possibile al corpo vettura che in questo tipo di auto fornisce il principale contributo nella generazione di downforce.

1.2.2 Aerodinamica odierna delle auto da corsa

Quindi esattamente come un aereo utilizza le ali per ottenere una spinta verso l'alto e volare, allo stesso modo è possibile installare una o più ali su un veicolo in modo da generare una spinta verso il basso (downforce) che aumenta la forza di attrito tra pneumatici e suolo e conseguentemente la velocità di percorrenza di una curva. Le prime applicazioni di ali su autoveicoli, come visto, erano abbastanza semplici (Figura 1.4), oggi si osservano geometrie molto più complesse caratterizzate da ali formate da molti elementi, questo permette di ottenere curvature molto più elevate dell'ala senza incorrere nella separazione dello strato limite con conseguente stallo dell'ala.



Figura 1.4: Ala anteriore Ferrari 312 B3 1974

L'aria, infatti, passa attraverso gli spazi che vengono lasciati tra i vari flap ed energizza lo strato limite sull'estradosso dei profili prevenendo separazione e stallo. Nelle configurazioni più estreme in Formula 1 si è arrivati anche a 6 flap in successione con deflessioni del flusso complessive che in alcune zone dell'ala arrivano quasi a 90°. In figura 1.5 è possibile osservare la Haas VF-18 che ha preso parte al Campionato del Mondo di Formula 1 del 2018, l'ultimo anno in cui è stato possibile sviluppare geometrie delle ali, e più in generale dell'aerodinamica delle vetture, molto complesse, prima di essere poi semplificate con un cambio di regolamento introdotto nel 2019. Si noti la complessità della geometria dell'ala anteriore e il numero di flap di cui è composta per i motivi appena spiegati.



Figura 1.5: Ala anteriore Haas VF-18 2018

1.3 Dallara Automobili S.p.A.

Giampaolo Dallara nasce nel 1936 a Varano De Melegari (PR). La passione per le auto e per le corse gli viene trasmessa dal padre e si iscrive quindi prima alla facoltà di ingegneria di Parma e dopo si trasferisce al Politecnico di Milano dove seguirà il corso di ingegneria aeronautica. Il Politecnico di Milano era allora dotato di una galleria del vento utilizzata anche da Ferrari per le sue prove ed è proprio qui che Dallara viene notato dagli ingegneri di Maranello che chiedono all'università di segnalare ad Enzo Ferrari lo studente. Nel 1959 quindi, a soli 23 anni, l'Ing. Dallara viene assunto da Ferrari per affiancare come assistente l'Ing. Carlo Chiti, allora direttore tecnico del reparto corse di Maranello. Dopo due anni si trasferisce in Maserati dove ha l'opportunità di lavorare al fianco di Giulio Alfieri e di collaborare con piloti del calibro di Roger Penske e Bruce McLaren. Nel 1963 entra a far parte di Lamborghini dove si occupa di vetture per uso stradale e prende parte al processo di creazione della Miura P400, la 350GT e l'Espada. Alla ricerca di un futuro agonistico si trasferisce poi alla De Tomaso dove sotto la guida di Frank Williams sviluppa prima una vettura di Formula 2 utilizzando per la prima volta tecnologie derivate dal settore aeronautico e, successivamente, una vettura di Formula 1. L'acquisizione di De Tomaso da parte di Ford porta alla fine del programma di corse e Dallara, a questo punto, decide di costruire autonomamente le proprie auto da corsa. Così, nel 1972, nasce Dallara Automobili da competizione, con sede nella casa di famiglia a Varano De Melegari.

Le prime vetture prodotte sono la sport prototipo VDM e la SP 1000 (figura 1.6 caratterizzata da un telaio in monoscocca di acciaio e tre posti con posizione di guida centrale, in modo da ottenere un perfetto bilanciamento della vettura pur rispettando il regolamento del campionato a cui prese parte, che prevedeva l'utilizzo di vetture con almeno due posti.

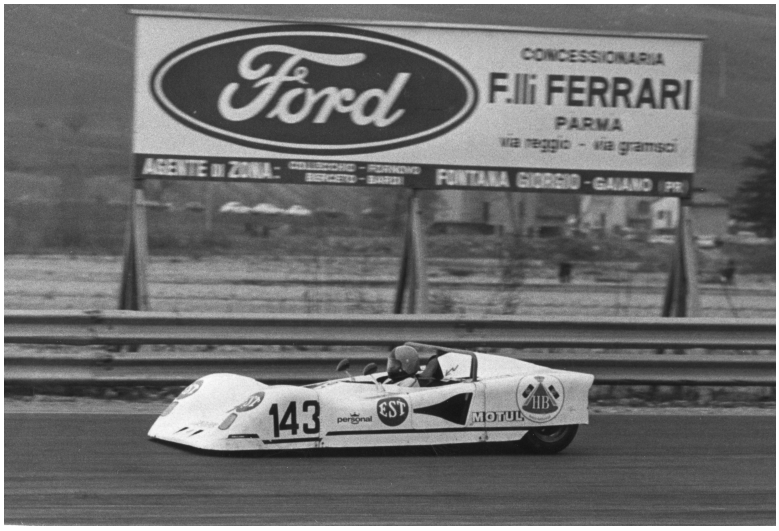


Figura 1.6: I° Trofeo Cinquetti 1973 - Dallara SP 1000

I prototipi dell'Ing. Dallara cominciano a guadagnare vittorie e questo insieme al successo della Dallara X1/9 richiama l'attenzione di Cesare Fiorio che contatta Dallara per creare una scuderia Lancia di vetture Sport, nel frattempo si iniziano a produrre le prime vetture di Formula 3 con aerodinamica sofisticata e utilizzo preponderante di fibra di carbonio, tecnologia fino ad allora riservata solo ad alcune auto di Formula 1. Nel 1991 viene inaugurato un nuovo stabilimento a Varano De Melegari e successivamente, nel 1998, Dallara ottiene la prima vittoria alla 500 miglia di Indianapolis, oggi Dallara è il costruttore che ha ottenuto più vittorie in questa corsa e nel 2012 viene aperto uno stabilimento proprio ad Indianapolis, dove vengono assemblate le Indycar, che da quell'anno in poi saranno progettate e costruite unicamente da Dallara. Questo avviene anche grazie agli sviluppi tecnologici in termini di prestazioni e sicurezza raggiunti dalle automobili prodotte da Dallara, molto più avanzati di quelli delle auto americane, grazie all'esperienza e specializzazione acquisita nella lavorazione della fibra di carbonio. Vi sono infatti due strutture dedicate a questo materiale: Dallara Compositi, polo produttivo, e il DARC (Dallara Advanced Composite Research Center), centro di ricerca e sperimentazione sui nuovi materiali. Oggi Dallara inoltre produce auto per tutti i principali campionati di corse del mondo quali Formula 2, Formula 3, Super Formula, Euroformula, Nascar e WEC (World Endurance Cup) [2] [3] [4].

Capitolo 2

Fluidodinamica Computazionale (CFD) e OpenFOAM

In questo capitolo verrà introdotto il concetto di Fluidodinamica Computazionale (CFD) e le informazioni riguardo al principale software per lo svolgimento del lavoro di tesi descritto in questa trattazione, OpenFOAM . Saranno inoltre presenti brevi cenni riguardo DAFoam, analizzato all'inizio del lavoro di ricerca, e sul perché è stato deciso di non utilizzarlo. La peculiarità principale di questi software è l'essere open source, questo rappresenta ovviamente un vantaggio sotto il punto di vista economico anche in vista di sviluppi futuri.

2.1 Computational Fluid Dynamics (CFD)

La Fluidodinamica Computazionale è uno strumento di simulazione inizialmente utilizzato in ambito aeronautico ed aerospaziale e successivamente esteso allo sviluppo di tecnologie di uso quotidiano, per esempio la produzione di energia, l'industria chimica e nucleare, l'industria marina e ultima ma non per importanza, per rimanere in tema con il seguente lavoro di tesi, l'industria dell'automotive.
La CFD è uno degli ultimi strumenti entrati a far parte di quella che viene denominata CAE (Computer Aided Engineering), unendosi a strumenti già esistenti come l'Analisi degli Elementi Finiti (FEA), ampiamente utilizzato per la meccanica dei solidi e lo studio delle vibrazioni, introdotto verso la metà degli anni '80. Il motivo di tale ritardo da parte della CFD è legato alla complessità delle equazioni in gioco quando si parla di fluidodinamica, prime tra tutte le equazioni di Navier-Stokes con le quali vengono risolti e modellati una serie di fenomeni di flusso quali flussi comprimibili e incomprimibili, laminari o turbolenti, monofase o multifase e tanti ancora.
Tra gli elementi utilizzati per sviluppare e implementare la CFD il ruolo principale lo ha sicuramente avuto il Metodo dei Volumi Finiti (FVM). Questo, è una tecnica numerica che trasforma le equazioni alle derivate parziali, che rappresentano le leggi di conservazione su volumi differenziali, in equazioni algebriche discrete su volumi finiti tendenzialmente denominati elementi o celle. Per fare questo il primo passo è la discretizzazione della geometria o del dominio di calcolo, questo viene quindi suddiviso in un determinato numero di elementi di forma arbitraria, ma non sovrapposti tra di loro, e su di essi vengono integrate le equazioni alle derivate parziali per ottenere delle equazioni algebriche. Si ottiene quindi un sistema di equazioni algebriche che viene risolto per ricavare le variabili dipendenti di ogni elemento.

Nel Metodo dei Volumi Finiti alcuni dei termini delle equazioni di conservazione vengono convertiti in flussi e valutati sulle facce delle celle, siccome il flusso che entra in un elemento deve essere uguale a quello che esce, ne consegue che l'FVM è un metodo strettamente conservativo e fa sì che sia particolarmente adatto all'applicazione alla CFD. Inoltre, nel Metodo dei Volumi Finiti è abbastanza facile implementare una grande quantità di condizioni al contorno in maniera non invasiva in quanto le variabili dipendenti vengono calcolate al centro delle celle e non sul contorno di queste, inoltre gli elementi possono andare a formare sia griglie di calcolo strutturate che non strutturate, utili nel caso di geometrie particolarmente complesse.
Tutte queste caratteristiche fanno si che l'FVM sia particolarmente adatto ad applicazioni quali lo studio dei flussi di fluido e quindi all'implementazione della CFD. Negli anni questa tecnica è stata ovviamente sviluppata e oggi siamo in grado di studiare moltissime applicazioni ed elaborare flussi incredibilmente complessi [5].

2.2 OpenFOAM

OpenFOAM (Open source Field Operation And Manipulation) è il principale software libero e open source per la Fluidodinamica Computazionale (CFD), di proprietà della OpenFOAM Foundation e distribuito esclusivamente sotto Licenza Pubblica Generale (GPL). La GPL offre agli utenti la libertà di modificare e ridistribuire il software e la garanzia di un uso continuo e gratuito.

2.2.1 Cenni storici

La prima versione di OpenFOAM è stato creato da Henry Weller nel 1989 con il nome di "FOAM". Successivamente nel dicembre del 2004 viene rilasciato da OpenCFD, la società fondata da Henry Weller, Chris Greenshields e Mattijs Janssens, "OpenFOAM" in versione open source. Da allora, nuove versioni di OpenFOAM sono state sviluppate, aggiornate e distribuite al pubblico con cadenza annuale. Nel 2011 OpenCFD viene inglobata in SGI Corp con l'obiettivo di offrire alle persone una soluzione CFD completamente integrata dove tutti i componenti software e hardware lavorano in sincronia. Per mantenere OpenFOAM un software open source tutte le parti hanno concordato nella realizzazione della OpenFOAM Foundation, con sede negli Stati Uniti d'America, in modo che OpenCFD potesse trasferirle marchio e copyright di OpenFOAM. Lo scopo della fondazione è stato quindi quello di distribuire OpenFOAM sotto licenza GPL al pubblico.

Nel 2012 SGI ha venduto OpenCFD a ESI Group, questa viene però lasciata due anni dopo da Henry Weller e Chris Greenshields per continuare lo sviluppo di OpenFOAM. A 10 anni di distanza dal primo lancio di OpenFOAM viene rilasciato OpenFOAM v2.3.1 che costituisce inoltre la base della prima introduzione online della linea di sviluppo di OpenFOAM: "OpenFOAM-dev".

Nello stesso anno la OpenFOAM Foundation viene trasferita nel Regno Unito modificandone la struttura di governance al fine di incentivare a finanziare la manutenzione di OpenFOAM, questo ha inoltre portato una maggiore collaborazione da parte di individui desiderosi di partecipare e contribuire allo sviluppo di OpenFOAM.

Nel marzo 2015 viene fondata CFD Direct da Weller, Greenshields e Jenya Collings con lo scopo di ereditare dalla OpenFOAM Foundation sviluppo e gestione di OpenFOAM. Con l'aiuto di un gruppo emergente di persone addette a manutenzione e test viene prodotto OpenFOAM v2.4.0 e dopo aver stipulato il Contributor Agreement per formalizzare i vari contributi allo sviluppo di OpenFOAM, firmato da alcuni collaboratori oltre che da Intel, blueCAPE e VTT Technical Research of Finland, il 3 novembre 2015 la fondazione rilascia OpenFOAM v3.0.0, la prima versione di OpenFOAM nata da OpenFOAM-dev, la linea di sviluppo pubblica.

Da quel momento in poi è stata lanciata ogni anno una nuova versione sviluppata e aggiornata di OpenFOAM fino ad oggi, dove si è arrivati a OpenFOAM v12 [6].

Esistono due versioni di OpenFOAM: la versione sopra descritta e la versione denominata ESI. Per svolvere il seguente lavoro di tesi sono state confrontate e utilizzate entrambe, Dallara preferisce l'utilizzo della versione di OpenFOAM ESI mentre per quanto riguarda la fase di apprendimento su come utilizzare il software e per lo svolgimento dei calcoli preliminari presso l'università è stata utilizzata la versione LTD. In ogni caso i risultati ottenuti con le due versioni di OpenFOAM sono equivalenti. Le uniche differenze apprezzabili in un primo momento sono legate al fatto che di OpenFOAM ESI viene rilasciata una nuova versione ogni 6 mesi, la versione utilizzata per il seguente lavoro di tesi è la v2412 (24 sta a significare anno 2024 e 12 sta a significare il mese di dicembre), mentre di OpenFOAM LTD viene rilasciata una versione ogni anno e quella utilizzata per la fase di apprendimento del software e i primi calcoli è stata la v10, rilasciata nel 2022. Inoltre, all'interno di OpenFOAM ESI è stato implementato un solutore dell'aggiunto continuo dal Politecnico di Atene (NTUA) a partire dalla versione v1912, per questo motivo è stato infine utilizzato OpenFOAM v2412 per svolgere i calcoli presenti all'interno di questo elaborato.

2.3 DAFoam

Il metodo dell'aggiunto è un approccio particolarmente efficiente che consente l'ottimizzazione basata sul gradiente di un sistema anche nel caso in cui questo sia caratterizzato da un gran numero di variabili di design. L'implementazione del metodo dell'aggiunto per un solver diretto basato su equazioni differenziali parziali, le equazioni di Navier-Stokes in fluidodinamica, è un compito che richiede molto tempo. Per ridurre il costo dell'implementazione del metodo dell'aggiunto è stato proposto un framework orientato agli oggetti (DAFoam) che permette di implementare il metodo dell'aggiunto basandosi su solutori diretti, come OpenFOAM, in regime stazionario, aggiungendo o modificando solo poche centinaia di righe di codice sorgente. Con questo approccio sono state ottenute un'eccellente velocità e scalabilità dell'aggiunto, con un massimo di 10 milioni di celle e 1536 core di CPU, e un errore medio nelle derivate delle equazioni aggiunte inferiore allo 0,1%. DAFoam è disponibile con licenza open-source ed è un potente strumento per l'ottimizzazione ad alta precisione e la progettazione progettazione di sistemi ingegneristici, come velivoli, veicoli terrestri, navi e turbomacchine [7].

L'idea iniziale per il seguente lavoro di tesi è stata quella di utilizzare questo software per il processo di ottimizzazione, tuttavia, lavorando con due griglie di calcolo rispettivamente da 12 e 16 milioni di celle non è stato possibile utilizzarlo a causa del costo computazionale troppo elevato che avrebbe comportato un processo di ottimizzazione.

2.4 Struttura di OpenFOAM

Per utilizzare correttamente OpenFOAM e studiare un caso come quello in esame è innanzitutto necessario scaricare e compilare OpenFOAM, nel nostro caso è stato compilato su un sistema operativo Linux, OpenSUSE. Per il funzionamento corretto di OpenFOAM sono poi necessarie per ogni caso tre directory principali:

- Directory ”0” o più in generale ”istante di tempo iniziale”
- Directory ”constant”
- Directory ”system”

All'interno di queste tre sono poi contenuti tutti i file di tipo ”.c” che contengono le informazioni necessarie allo svolgimento del calcolo.

2.4.1 Directory ”0”

La directory ”0” comprende al suo interno tutte le informazioni relative alle condizioni al contorno dei campi di velocità, pressione e grandezze relative alla turbolenza. Nel nostro caso all'interno della directory ”0”, considerando come esempio un caso in cui si utilizza il modello di turbolenza k-ω SST è possibile trovare:

- ”U”: Velocità.
- ”p”: Pressione.
- ”k”: Energia cinetica turbolenta.
- ”omega”: Tasso di dissipazione specifico della turbolenza.
- ”nut”: Viscosità cinematica turbolenta.

Al suo interno ogni file è poi suddiviso e fornisce informazioni sulle condizioni al contorno su tutte le patch del dominio di calcolo. Sarà possibile trovare maggiori dettagli nel Capitolo 4 relativo al setup di calcolo.

2.4.2 Directory ”constant”

La directory ”constant” contiene tutte le informazioni riguardo le proprietà fisiche del caso in esame e le informazioni riguardanti la griglia di calcolo (mesh) in una subdirectory denominata ”polyMesh”. Troviamo quindi al suo interno:

- Directory ”polyMesh”.
- ”transportProperties”: valore della viscosità cinematica ν per il caso in esame.
- ”turbulenceProperties”: informazioni a proposito del modello di turbolenza utilizzato.

All’interno della directory ”polyMesh” sono invece contenuti i seguenti file:

- ”boundary”: elenco delle patch del dominio con informazioni specifiche per ognuna.
- ”faces”: elenco delle facce degli elementi della griglia di calcolo.
- ”neighbour”: elenco delle etichette delle celle vicine.
- ”owner”: elenco di etichette delle celle.
- ”points”: elenco di vettori che descrivono i vertici delle celle.

Anche in questo caso sarà possibile trovare maggiori dettagli nel Capitolo 4.

2.4.3 Directory ”system”

Contiene tutte le informazioni e i parametri necessari al vero e proprio svolgimento del calcolo. Nel nostro caso all’interno di questa directory troviamo:

- ”controlDict”: contiene i parametri di controllo del calcolo, come istante di tempo iniziale e finale, e le funzioni utili ad esempio per il calcolo di grandezze quali coefficienti di lift e drag o valori massimi e minimi dei campi di velocità e pressione.
- ”fvSchemes”: contiene le informazioni a proposito degli schemi di discretizzazione utilizzati durante il calcolo.
- ”fvSolution”: contiene le informazioni riguardanti i solutori delle equazioni, i fattori di rilassamento e altri parametri di controllo per gli algoritmi utilizzati durante il calcolo.
- ”checkMeshDict”: contiene la informazioni relative al check della mesh per verificare che tutte le caratteristiche di questa siano accettabili al fine di ottenere dei buoni risultati dal calcolo.
- ”decomposeParDict”: permette di effettuare il calcolo in parallelo su più core, al suo interno sono contenute le informazioni necessarie per una corretta ripartizione della mesh.

Nel Capitolo 4 sarà possibile visualizzare i dettagli a proposito di ”controlDict”, ”fvSchemes” e ”fvSolution” che sono i tre file più importanti ed essenziali all’interno di questa directory.

Capitolo 3

Equazioni governanti

All'interno di questo capitolo verranno introdotti i concetti relativi alle equazioni che governano il comportamento di un flusso di fluido, le equazioni di Navier-Stokes. Verranno inoltre fornite informazioni sui modelli di turbolenza in generale e più nello specifico su quelli utilizzati per il seguente lavoro di tesi: $k-\omega$ SST e Spalart-Allmaras.

3.1 Equazioni di Navier-Stokes

Le equazioni di Navier-Stokes esprimono il legame all'interno di un flusso di fluido tra velocità, pressione, temperatura e densità. Si compongono di un sistema di equazioni differenziali che sono troppo complesse per essere risolte analiticamente, tuttavia, è possibile introducendo un piccolo errore risolverle numericamente. Queste equazioni (3.1) si compongono rispettivamente di:

- Equazione di conservazione della massa
- Equazione di conservazione della quantità di moto
- Equazione di conservazione dell'energia

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \\ \dfrac{\partial (\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \nabla \cdot \tau + S_u \\ \dfrac{\partial (\rho e_t)}{\partial t} + \nabla \cdot (\rho e_t \vec{u}) = \nabla \cdot q - \nabla \cdot (p \vec{u}) + \tau : \nabla \vec{u} + S_e \end{cases} \tag{3.1}$$

Se si considera una casistica come quella analizzata all'interno del seguente elaborato, dove non viene tenuto in considerazione lo scambio termico e il flusso è caratterizzato da un numero di Mach inferiore a 0.3, quindi è incomprimibile, si ottiene la forma semplificata di queste equazioni (3.2):

$$\begin{cases} \nabla \cdot \vec{u} = 0 \\ \rho \dfrac{D\vec{u}}{Dt} = -\vec{\nabla} p + \mu \nabla^2 \vec{u} \end{cases} \tag{3.2}$$

3.2 Modelli di turbolenza

Nello studio dei flussi turbolenti l'obiettivo principale è ottenere una teoria o un modello quantitativo trattabile che possa essere utilizzato per calcolare quantità di interesse. Un secolo di esperienza ha dimostrato che il "problema della turbolenza" è notoriamente difficile e al momento non ci si aspetta di trovare una semplice teoria analitica che lo risolva. Invece, si spera di utilizzare la potenza computazionale sempre crescente dei computer odierni per calcolarne le proprietà rilevanti.

Il problema principale che impedisce lo sviluppo di una semplice teoria analitica è legato al fatto che il campo di velocità in un flusso turbolento è tridimensionale, dipendente dal tempo e casuale. I vortici più grandi sono quasi di pari dimensione rispetto alla larghezza caratteristica del flusso e, di conseguenza, sono direttamente influenzati dalla geometria, ad esempio delle pareti solide. Non è quindi possibile trovare uno standard che racchiuda le varie situazioni.

In una simulazione di flusso turbolento, le equazioni vengono risolte per un campo di velocità dipendente dal tempo. Al contrario, in un modello di turbolenza, le equazioni vengono risolte per alcune quantità medie [8].

Gli approcci utilizzati per la modellazione del flusso turbolento sono molteplici, i principali sono la DNS (Direct Numerical Simulation), la LES (Large Eddy Simulation), i modelli RANS (Reynolds Average Navier-Stokes) e i modelli ibridi che sono una combinazione di quelli appena citati.

La DNS consiste nel risolvere direttamente l'intero spettro di scale turbolente utilizzando un approccio noto come simulazione numerica diretta. In questo caso non è richiesta alcuna modellazione, tuttavia, un tipo di approccio del genere non è utilizzabile per problemi industriali che coinvolgono flussi con numeri di Reynolds molto elevati. Il costo richiesto per risolvere l'intero intervallo di scale con una DNS è proporzionale a Re_t^3, dove Re_t è il numero di Reynolds turbolento. Chiaramente, per numeri di Re_t elevati, il costo computazionale diventa insostenibile anche nel caso di computer molto potenti.

La LES permette di risolvere i grandi vortici attraverso una DNS mentre quelli piccoli vengono modellati. Risolvendo solo i grandi vortici è possibile utilizzare mesh molto più rade e time-step di dimensioni maggiori in una LES rispetto ad una DNS. Tuttavia, una LES richiede ancora griglie di calcolo sostanzialmente più fini di quelle tipicamente utilizzate per i calcoli RANS. Inoltre, i calcoli devono essere eseguiti per un tempo sufficientemente lungo per ottenere statistiche stabili del flusso modellato. Di conseguenza, il costo computazionale, in termini di memoria (RAM) e tempo CPU, di una LES è maggiore di quello che si ha per un calcolo RANS, quindi, è necessario specialmente per applicazioni industriali utilizzare procedure di calcolo ad alte prestazioni, ad esempio il calcolo in parallelo su più CPU. [9].

3.2.1 Modelli RANS

Per risolvere il problema legato all'enorme costo computazionale di DNS e LES è possibile andare a mediare nel tempo le equazioni di Navier-Stokes in forma incomprimibile scritte precedentemente (3.2). La prima equazione esprime la continuità, la seconda la conservazione della quantità di moto. Per farlo è necessario sostituire all'interno di (3.2) velocità e pressione scomposte nelle loro componenti medie e fluttuazioni come visibile in (3.3).

$$\begin{aligned} \vec{u}(x,t) &= U(x) + \vec{u}'(x,t) \\ p(x,t) &= P + p'(x,t) \end{aligned} \tag{3.3}$$

Mediando le equazioni ottenute è possibile eliminare la componente relativa alle fluttuazioni di velocità e pressione, in questo modo si alleggerisce il calcolo a scapito però ovviamente dell'accuratezza dei risultati. Le equazioni che si ottengono (3.4) vengono chiamate RANS equations, dove RANS sta per Reynolds Average Navier-Stokes.

$$\begin{cases} \nabla \cdot U = 0 \\ \rho \dfrac{DU}{Dt} = -\vec{\nabla} P + \mu \nabla^2 U + \nabla \tau_R \end{cases} \tag{3.4}$$

Dove τ_R (3.5) indica il Tensore degli sforzi di Reynolds e può essere riscritto come:

$$\tau_R = -\rho(\overline{\vec{u}'\vec{u}'}) = \rho \begin{pmatrix} (\overline{u'^2}) & -(\overline{u'v'}) & -(\overline{u'w'}) \\ -(\overline{u'v'}) & (\overline{v'^2}) & -(\overline{v'w'}) \\ -(\overline{u'w'}) & -(\overline{v'w'}) & (\overline{w'^2}) \end{pmatrix} \tag{3.5}$$

Il problema che ne consegue però è che, mediando le equazioni come descritto sopra, cioè decomponendo velocità e pressione, introduciamo sempre delle incognite aggiuntive, cioè quelle relative al tensore degli sforzi di Reynolds. Inoltre, compaiono momenti di ordine superiore delle fluttuazioni di velocità. Per esempio se volessimo risolvere le equazioni del trasporto di un momento di ordine n-esimo troveremmo all'interno delle equazioni un momento di ordine superiore ($n+1$-esimo) e così via. Questo richiede che, ad un certo punto, il ciclo debba essere chiuso andando a modellare alcuni termini, più è elevato l'ordine che è stato raggiunto più saranno numerosi i termini da modellare. A seconda del valore che assume il valore n, quindi, gli approcci sono differenti:

- n=1: Si utilizzano modelli basati sulla viscosità turbolenta.
- n=2: Si utilizza il modello di turbolenza RSM che verrà successivamente descritto.

Per i flussi tridimensionali in generale abbiamo quattro grandezze incognite che sono la pressione e le tre componenti della velocità. Queste, insieme alle sei incognite aggiuntive introdotte da τ_R ci portano ad avere in totale 10 incognite con sole quattro equazioni a disposizione, una per la conservazione della massa e tre per la conservazione della quantità di moto (3.4) [10]. A questo punto per risolvere questo problema andiamo a descrivere più nel dettaglio gli approcci sopra introdotti, sarà infatti possibile:

- Utilizzare l'ipotesi di viscosità turbolenta introdotta da Boussinesq nel 1877. Questa, per un fluido newtoniano, è matematicamente analoga alla relazione sforzo-velocità di deformazione. Secondo l'ipotesi il tensore della velocità di deformazione mediato $\overline{S}_{ij}$ è proporzionale a $(-\rho(\overline{\vec{u}'\vec{u}'}) + \frac{2}{3}\rho k\mathbf{I})$ nel seguente modo:

$$\rho(\overline{\vec{u}'\vec{u}'}) = \frac{1}{3}\rho(2k)\mathbf{I} - \frac{1}{3}\rho(2k)\mathbf{I} + \rho(\overline{\vec{u}'\vec{u}'}) \tag{3.6}$$

  Dove "k" è l'energia cinetica turbolenta e "**I**" la matrice identità. Raccogliendo un termine $2k$ si ottiene:

$$\rho(\overline{\vec{u}'\vec{u}'}) = 2k(\frac{1}{3}\rho\mathbf{I} + \frac{\rho(\overline{\vec{u}'\vec{u}'})}{2k} - \frac{1}{3}\rho\mathbf{I}) \tag{3.7}$$

  A questo punto introduciamo il termine $\mathbf{b}_{ij}$, secondo l'ipotesi di Boussinesq otteniamo:

$$\rho\mathbf{b}_{ij} = \frac{\rho(\overline{\vec{u}'\vec{u}'})}{2k} - \frac{1}{3}\rho\mathbf{I} = -\frac{\mu_t}{k}\overline{S}_{ij} \tag{3.8}$$

  Dove μ_t è la viscosità turbolenta. Ricordando che $2\overline{S}_{ij}$ può essere scritto come $(\nabla U + \nabla U^T)$ e combinando (3.7) e (3.8) si ottiene il legame tra il tensore degli sforzi di Reynolds e il tensore delle deformazioni attraverso la viscosità turbolenta.

$$-\rho(\overline{\vec{u}'\vec{u}'}) = \mu_t(\nabla U + \nabla U^T) - \frac{2}{3}\rho k\mathbf{I} \tag{3.9}$$

  A questo punto è possibile scrivere le RANS in forma chiusa introducendo la viscosità effettiva come somma di viscosità e viscosità turbolenta:

$$\begin{cases} \nabla \cdot U = 0 \\ \rho\dfrac{DU}{Dt} = -\nabla(P + \dfrac{2}{3}\rho k) + \nabla\mu_{eff}(\nabla U + \nabla U^T) \end{cases} \tag{3.10}$$

  con:

$$\mu_{eff}(x,t) = \mu + \mu_t(x,t) \tag{3.11}$$

  Il problema a questo punto rimane unicamente ottenere "μ_T" e "k".

- Scrivere un'equazione risolutiva per ognuna delle componenti di "τ_R" ma questo comporta un più grande costo computazionale. Questo tipo di approccio si traduce nel modello "RSM" che è infatti molto complicato da portare a convergenza soprattutto nel caso in cui non vengano analizzate geometrie semplici oppure si abbiano griglie di calcolo molto fini.

Per ricavare "μ_t" si possono utilizzare diversi modelli di turbolenza che si differenziano per il numero di equazioni differenziali introdotte per determinare la viscosità turbolenta. Vi sono i modelli "zero equation" che utilizzano un'equazione algebrica per determinare "μ_t", "one equation" che utilizzano un'equazione differenziale per determinare "μ_t" e "two equations" che utilizzano due equazioni differenziali per determinare "μ_t". I modelli "one equation" e "two equations" sono quelli oggetto della validazione argomento della seguente trattazione, a seguire è quindi possibile trovare un analisi di questi più dettagliata.

Modello $k - \omega$ SST

Il modello $k - \omega$ SST (Shear Stress Transport) deriva dal modello $k - \omega$ standard, si propone però di eliminare uno dei punti deboli di questo modello, vale a dire la sensibilità alle condizioni al contorno del flusso libero che può avere un effetto significativo sulla soluzione. Nel corso degli anni al modello $k - \omega$ sono stati aggiunti termini di produzione alle equazioni sia di "k" che di "ω". Questo, ha comportato un miglioramento nell'accuratezza del modello nella previsione del comportamento del flusso libero. Per il seguente elaborato è stato scelto di utilizzare il modello $k - \omega$ SST in quanto oltre ad essere il più evoluto ha la peculiarità di essere "$y+$ insensitive", questo significa che funziona correttamente e produce risultati corretti sia che si utilizzi una griglia di calcolo caratterizzata da un boundary layer con "first layer thickness" elevato e quindi di tipo "wall modeling" dove il flusso non viene risolto fino a parete ma bensì modellato, sia che si utilizzi una griglia di calcolo con un "boudary layer" caratterizzato da un "first layer thickness" ridotto e quindi "wall resolving" dove il flusso viene risolto fino a parete. Riassumendo è quindi possibile ottenere dei buoni risultati sia utilizzando una "mesh" caratterizzata da un valore di "$y+$" elevato (tra 30 e 300) sia utilizzandone una caratterizzata da un valore di "$y+$" ridotto (tra 0 e 10). Il modello $k - \omega$ SST si basa su due equazioni: una per l'energia cinetica della turbolenza "k" (3.12) e il tasso di dissipazione specifico "ω" (3.13) [11]:

$$\frac{D(\rho k)}{Dt} = \tilde{P}_k - \rho\beta^*\omega k + \nabla[(\mu + \sigma_k\mu_T)\nabla k] \tag{3.12}$$

$$\frac{D(\rho\omega)}{Dt} = \alpha\rho S^2 - \rho\beta\omega^2 - \nabla[(\mu + \sigma_\omega\mu_t)\nabla\omega] + 2\rho\sigma_{\omega 2}(1 - F_1)\frac{1}{\omega}\nabla k\nabla\omega \tag{3.13}$$

All'interno di queste equazioni è importante notare la presenza del production limiter "$\tilde{P}_k$", per evitare l'eccessiva produzione di turbolenza nei punti di ristagno, definito come:

$$P_k = \mu_t\frac{\partial\vec{u_i}}{\partial x_j}(\frac{\partial\vec{u_i}}{\partial x_j} + \frac{\partial\vec{u_j}}{\partial x_i}) \tag{3.14}$$

$$\tilde{P}_k = min(P_k, 10\beta^*\rho k\omega) \tag{3.15}$$

e del valore "F_1" che controlla il comportamento del modello di turbolenza in quanto assume valore 0 oppure 1 a seconda che ci si trovi all'esterno o all'interno dello strato limite. La sua formulazione è:

$$F_1 = tanh\{\{min[max(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}), \frac{4\rho\sigma_{\omega 2}k}{CD_{k\omega}y^2}]\}^4\} \tag{3.16}$$

La viscosità turbolenta "ν_t" viene ottenuta attraverso l'equazione 3.14:

$$\nu_t = \frac{a_1 k}{max(a_1\omega, SF_2)} \tag{3.17}$$

Dove "S" è la misura invariante del tasso di deformazione e "F_2" viene calcolato attraverso:

$$F_2 = tanh\{[max(\frac{2\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega})]^2\} \tag{3.18}$$

i coefficienti inseriti e utilizzati all'interno di questo modello in OpenFOAM sono stati riportati nella tabella sottostante 3.1.

α_{k1}	α_{k2}	$\alpha_{\omega1}$	$\alpha_{\omega2}$	β_1	β_2	γ_1	γ_2	β^*	a_1	b_1	c_1
0.850	1.000	0.500	0.856	0.075	0.0828	0.556	0.440	0.090	0.310	1.000	10.00

Tabella 3.1: Costanti del modello $k-\omega$ SST

Modello Spalart-Allmaras

A causa di problematiche durante lo svolgimento della fase di calcolo del seguente lavoro di tesi è stato necessario ricercare una soluzione più stabile rispetto a quella offerta dal modello $k-\omega$ SST, per ottenere questi risultati è stato utilizzato il modello Spalart-Allmaras. Si tratta di un modello ad una equazione basato sulla viscosità turbolenta modificata "$\tilde{\nu}$" che è stato inizialmente progettato e pensato specificamente per le applicazioni aeronautiche. Inoltre, è stato possibile osservare come fornisca buoni risultati quando si studiano strati limite soggetti a gradienti avversi di pressione. In origine per far si che questo modello funzionasse correttamente era necessario utilizzare mesh che risolvessero lo strato limite fino in prossimità della parete ($y+ \sim 1$), oggi il modello è stato migliorato e permette di ottenere buoni risultati anche con mesh più grossolane con valori di "$y+$" fino a 30. Scriviamo quindi le equazioni introdotte dal modello di Spalart-Allmaras standard che permettono di determinare la viscosità turbolenta per chiudere definitivamente le equazioni RANS [12].

La viscosità turbolenta viene calcolata attraverso:

$$\nu_t = \tilde{\nu} f_{v1} \tag{3.19}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \tag{3.20}$$

$$\chi = \frac{\tilde{\nu}}{\nu} \tag{3.21}$$

e la viscosità turbolenta modificata "$\tilde{\nu}$" viene ricavata con l'equazione:

$$\frac{D\tilde{\nu}}{Dt} = C_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} + \frac{1}{\sigma}[\nabla((\nu + \tilde{\nu})\nabla\tilde{\nu}) + C_{b2}(\nabla\tilde{\nu})^2] - (C_{w1} - \frac{C_{b1}}{k^2} f_{t2})(\frac{\tilde{\nu}}{d})^2 + f_{t1}\Delta\vec{u}^2 \tag{3.22}$$

Dove "$\tilde{S}$" e "f_{v2}" vengono calcolati rispettivamente come:

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \tag{3.23}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{3.24}$$

con "S" e "d" pari rispettivamente alla vorticità e alla distanza dalla parete più vicina.

Rimangono ora da ricavare le tre funzioni "f_w", "f_{t1}" e "f_{t2}" calcolate rispettivamente come:

$$f_w = g\left[\frac{1+C_{w3}^6}{g^6+C_{w3}^6}\right]^{\frac{1}{6}} \tag{3.25}$$

$$f_{t2} = C_{t3}e^{(-C_{t4}\chi^2)} \tag{3.26}$$

$$f_{t1} = C_{t1}g_t e^{(-C_{t2}\frac{\omega_t^2}{\Delta\vec{u}^2}[d^2+g_t^2 d_t^2])} \tag{3.27}$$

con "r" e "g" che vengono ottenute rispettivamente con:

$$r \equiv \frac{\tilde{\nu}}{\tilde{S}k^2d^2} \tag{3.28}$$

$$g = r + C_{w2}(r^6 - r) \tag{3.29}$$

Infine, nella tabella sottostante 3.2 vengono mostrati tutti i valori dei coefficienti sopra citati inseriti in OpenFOAM, si osservi che $C_{w1} = f(C_{b1}, C_{b2}, \sigma, k)$ ma in questo caso è stato riportato il valore numerico calcolato in funzione dei coefficienti di valore noto.

C_{b1}	σ	C_{b2}	k	C_{w1}	C_{w2}	C_{w3}	C_{v1}	C_{t1}	C_{t2}	C_{t3}	C_{t4}
0.136	0.667	0.622	0.410	2.763	0.300	2.000	7.100	1.000	2.000	1.200	0.500

Tabella 3.2: Coefficienti del modello di turbolenza Spalart-Allmaras

Capitolo 4

Geometria, dominio di calcolo, ”mesh” e condizioni al contorno

All'interno di questo capitolo verranno descritti tutti gli elementi che caratterizzano il dominio di calcolo, la geometria iniziale analizzata e le condizioni al contorno associate a ciascuno di questi elementi. Verranno inoltre esaminate le due griglie di calcolo utilizzate per effettuare le simulazioni e sottolineate le differenze tra queste.

4.1 Geometria, ”patch” dell'auto e dominio di calcolo

La geometria iniziale è stata fornita da Dallara ed è una generica geometria di una vettura a ruote scoperte tipo formula. Si tratta di una geometria semplificata in quanto l'obiettivo principale della trattazione non è l'analisi della geometria iniziale bensì l'ottimizzazione di questa, in particolare dell'ala. Per questo motivo non vi sono elementi della sospensione e impianto frenante, il cerchione della ruota contrariamente alla realtà non è cavo ma pieno e la posizione della ruota è stata modificata arbitrariamente, non in modo eccessivo, per cercare di ottenere un risultato dalle simulazioni il più stazionario possibile, per non incorre in problemi quando si andrà ad implementare il metodo dell'aggiunto. La ruota è quindi considerata unicamente come elemento di disturbo per l'ala e non fa parte della procedura di ottimizzazione.
L'ala, d'altro canto, essendo la protagonista dell'ottimizzazione è più dettagliata. Questa, è composta da un ”flap” principale (”main flap”) collegato alla vettura attraverso un pilone di sostegno. Al ”main flap” è collegato l'”endplate” che oltre ad avere un suo preciso scopo sotto il punto di vista aerodinamico ha anche il ruolo di sostenere i ”flap” inferiore e superiore, denominati rispettivamente ”lower flap” e ”upper flap”.

Di seguito è possibile osservare in figura 4.1 una rappresentazione della geometria utilizzata evidenziando le varie sezioni (”patch”) in cui è diviso il file ”.stl” originale che verranno nominate successivamente.

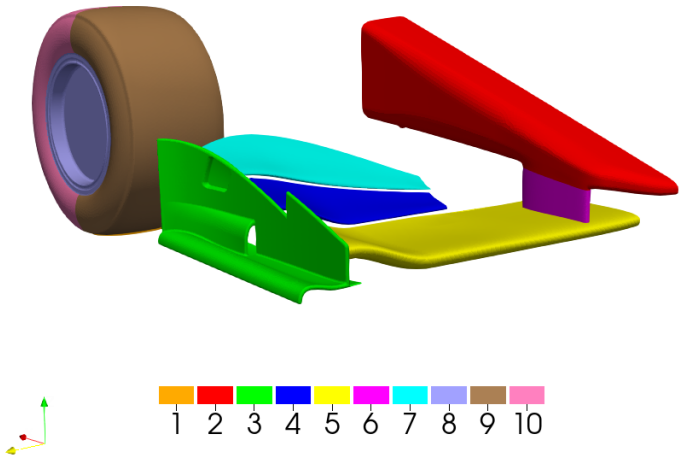


Figura 4.1: Geometria iniziale divisa in singole patch

Di seguito è possibile osservare in tabella 4.2 i nomi associati alle singole patch che verranno poi ritrovati all’interno della descrizione del setup di OpenFOAM.

N° Patch	Nome patch
1	w00np-fwl
2	w01-bw-nose
3	w02-fw-ep
4	w02-fw-lower-flap
5	w02-fw-main
6	w02-fw-pylon
7	w02-fw-upper-flap
8	w06b-fwlr-rim
9	w06b-fwlr-tyre-front
10	w06b-fwlr-tyre-rear

Tabella 4.1: Nomenclatura delle patch della geometria iniziale

La patch N° 1, ”w00np-fwl”, poco visibile in figura, è una sottile fascia che raccorda lo pneumatico al terreno, questo viene fatto in quanto altrimenti l’angolo compreso tra ruota e terreno sarebbe troppo acuto e comporterebbe una ”skewness” molto elevata delle celle in quella zona.

Allo stesso modo nella figura 4.2 è possibile osservare il dominio di calcolo utilizzato diviso in singole ”patch” e in tabella **??** è riportata la nomenclatura di queste, utile anche in questo caso quando verranno analizzati i setup relativi alle condizioni al contorno sul dominio.

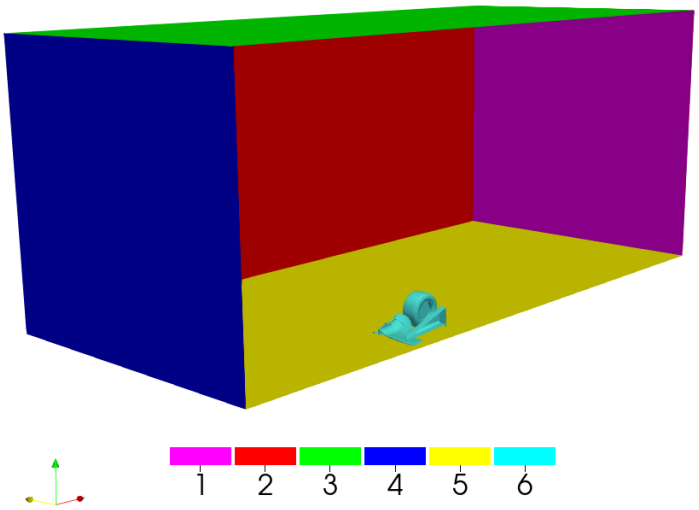


Figura 4.2: Dominio di calcolo diviso in singole patch

N° Patch	Nome patch
1	p-outlet
2	s-domain-limit-lat
3	s-domain-limit-sup
4	v-inlet
5	w00-ground
6	Car

Tabella 4.2: Nomenclatura delle patch del dominio

Il dominio è aperto da un lato per permettere la facile visualizzazione delle varie parti, la parete, che è stata cancellata per permettere una migliore visualizzazione del dominio, è comunque da ritenersi uguale in tutto e per tutto a quella al punto 2, fatta eccezione ovviamente per la presenza dell’auto adiacente a questa.

4.2 Griglie di calcolo (mesh)

Le ”mesh” utilizzate per svolgere i calcoli necessari per sviluppare il seguente elaborato di tesi sono state generate e fornite da Dallara. Si tratta di ”mesh” per la maggior parte di tipo strutturato formate quindi da esaedri, una griglia di calcolo di questo tipo è particolarmente indicata per i casi in cui è nota la direzione principale e prevalente lungo la quale si muove il flusso di fluido, l'insieme delle celle tende in un certo senso a replicare le linee di corrente permettendo così uno studio più accurato dei flussi. Le ”mesh” presentano degli infittimenti nelle zone ad elevata curvatura della geometria, come ”leading edge” e ”trailing edge” dei flap, e nella zona di contatto tra terreno e ruota della vettura. Per aumentare la precisione del calcolo sono stati inseriti 3 corpi di influenza concentrici che vanno ad infittire la griglia, riducendo le dimensioni globali degli elementi al loro interno, a mano a mano che ci si avvicina alla vettura come visibile in Figura 4.3.

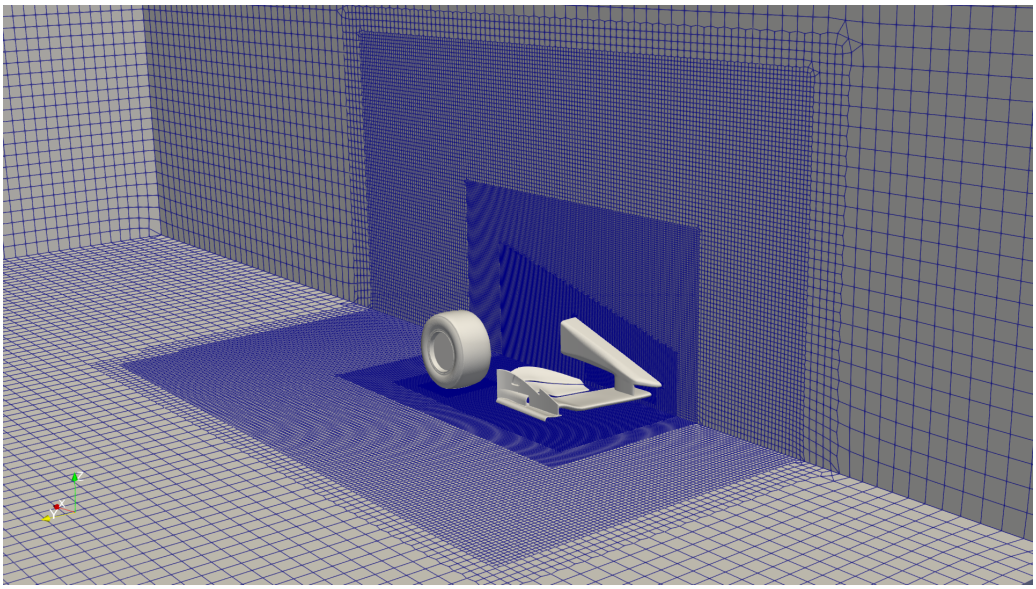


Figura 4.3: Dettaglio dei corpi di influenza e relativi infittimenti

La differenza principale tra le due ”mesh” risiede all'interno del ”boundary layer”, viene generato per studiare più dettagliatamente il flusso sulle superfici sulle quali si ha la formazione di uno strato limite, legato alla viscosità non nulla del fluido evolvente attorno alla geometria, condizione che fa si che, a contatto con la parete, il fluido abbia la velocità della parete stessa. Uno studio di questo tipo è particolarmente importante quando si studia l'aerodinamica di un corpo solido in quanto l'evolvere dello strato limite influenza fortemente le prestazioni dei flap o dei profili e deve quindi essere eseguito correttamente per ritenere i risultati ottenuti come validi. Il ”boundary layer” generato all'interno della griglia di calcolo è una zona nella quale le celle vengono create seguendo delle direttrici perpendicolari alla parete a cui si riferiscono e hanno uno spessore crescente allontanandosi da questa. Esistono diversi vincoli per la generazione del ”boundary layer”, tendenzialmente si calcola il ”first layer thickness” basandosi sul valore di ”$y+$” desiderato e si definisce un numero di strati e un andamento dell'accrescimento delle dimensioni della cella in modo da raccordare in modo adeguato il ”boundary layer” con il resto della celle circostanti.

Si otterranno quindi due ”boundary layer” differenti a seconda del valore di ”$y+$” ricercato e quindi due differenti griglie di calcolo:

- Una caratterizzata da un valore elevato di ”$y+$” (wall modeling) da circa 12 milioni di elementi.
- Una caratterizzata da un valore ridotto di ”$y+$” (wall resolving) da circa 16 milioni di elementi.

Queste verranno successivamente distinte attraverso i termini ”hyp” (High Y Plus) e ”lyp” (Low Y Plus).
In Figura 4.4 è possibile osservare un dettaglio riguardante il ”boundary layer”, ottenuto in fase di generazione della griglia, per ottenere un valore elevato di ”$y+$” e allo stesso modo in Figura 4.5 è possibile notare invece le caratteristiche di uno adatto alla risoluzione del flusso fino in prossimità della parete, quindi caratterizzato da un valore di ”$y+$” ridotto.

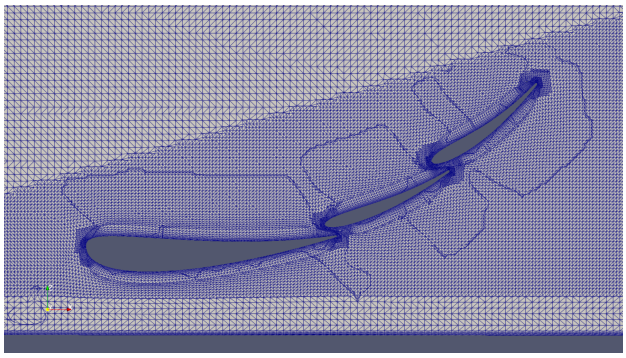


Figura 4.4: Boundary layer mesh hyp

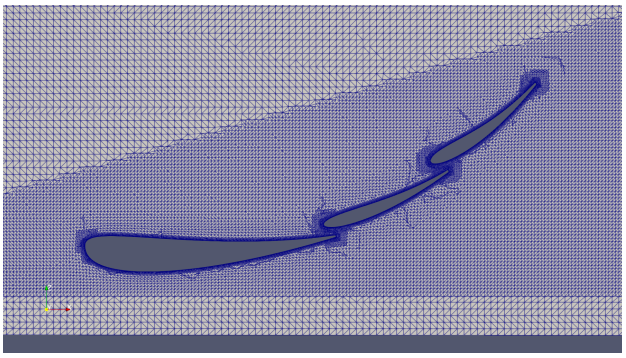


Figura 4.5: Boundary layer mesh lyp

Di seguito è possibile osservare il valore di ”$y+$” ottenuto nei due casi in Figura 4.6 e 4.7 dove vengono mostrati i contour di ”$y+$” su auto, ruota e terreno, importante notare la differenza di valori nelle scale dei due contour.

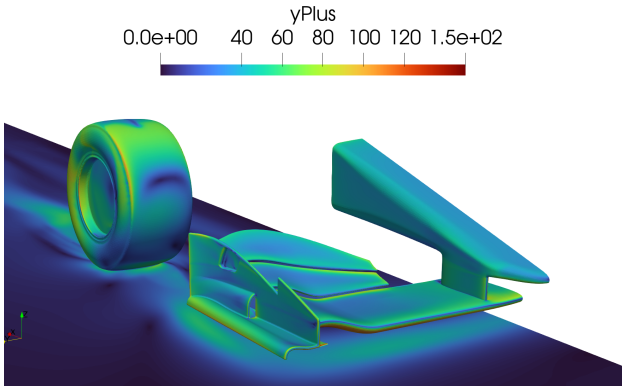


Figura 4.6: Contour y+ mesh hyp

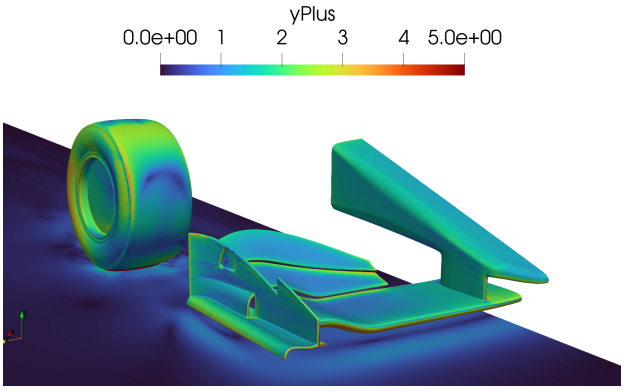


Figura 4.7: Contour y+ mesh lyp

4.2.1 Determinazione del ”first layer thickness”

Per generare un boudary layer che permetta di ottenere il valore di ”$y+$” desiderato è necessario conoscere il ”first layer thickness”, vale a dire l'altezza della prima cella a contatto con la parete della geometria.
Esistono dei programmi online per generare questo valore a partire da una serie di dati in input [13], in particolare se si considera come esempio un profilo alare servono:

- Velocità del flusso indisturbato (u)
- Densità del fluido (ρ)
- Viscosità dinamica (μ)
- Lunghezza caratteristica del sistema, nel nostro caso la corda del profilo (c)
- Valore desiderato di ”$y+$”

Viene calcolata la velocità ”u_τ” detta ”friction velocity” come funzione dello sforzo tangenziale a parete ”τ_w” (3.1):

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{4.1}$$

La viscosità cinematica può essere riscritta come rapporto tra la viscosità dinamica e la densità (3.2):

$$\nu = \frac{\mu}{\rho} \tag{4.2}$$

A questo punto possiamo scrivere la formulazione per ”$y+$”:

$$y+ = \frac{y \cdot u_\tau}{\nu} \tag{4.3}$$

Dalla formulazione di ”$y+$”, rielaborandola, è possibile ricavare il valore di ”y” ossia il ”First layer thickness” ricercato da inserire al momento della generazione della griglia di calcolo.

4.3 Condizioni al contorno

All'interno di questo sottocapitolo, come anticipato nel Capitolo 2, verranno descritte le condizioni al contorno e le condizioni iniziali del calcolo, inoltre, verrà descritto come impostare la struttura di un caso di OpenFOAM in modo da garantire il corretto funzionamento dei calcoli e l'ottenimento di buoni risultati [6].

4.3.1 Directory "0"

Come descritto nel Capitolo 2 all'interno di questa directory sono contenute tutte le informazioni relative ai campi di pressione, velocità e grandezze fisiche della turbolenza, all'istante iniziale del calcolo. All'interno di questa cartella è possibile trovare un file di tipo ".c" per ogni grandezza sopra elencata con all'interno delle informazioni, scritte seguendo le regole del linguaggio di programmazione C++, utili per il corretto funzionamento di OpenFOAM e dello svolgimento del calcolo. All'interno dei file della directory "0" in particolare è possibile trovare diversi "dictionary", sottogruppi contenenti le informazioni necessarie al calcolo:

- FoamFile: fornisce informazioni a proposito del documento stesso come versione del file, formato (binario o ASCII), classe (se si tratta di una grandezza scalare o vettoriale), posizione del file e oggetto del file (velocità, pressione, ecc.)
- dimensions: fornisce informazioni a proposito dell'unità di misura della grandezza contenuta nel file attraverso un formato tipo: [0 0 0 0 0 0 0] dove la posizione del numero nella quadra indica rispettivamente:
  [**kg**(massa) **m**(distanza) **s**(tempo) **K**(temperatura) **mol**(quantità) **A**(corrente) **cd**(luminosità)]
  mentre il valore inserito indica l'esponente associato ad ogni grandezza.
  Per esempio: [0 1 -1 0 0 0 0] equivale all'unità di misura $\frac{m}{s}$.
- internalField: contiene la condizione iniziale imposta su tutto il volume interno al dominio di calcolo
- boundaryField: contiene la descrizione delle condizioni al contorno per ogni singola patch del dominio. Per ogni patch va poi definito il tipo di condizione al contorno, poi, a seconda della tipologia che è stata scelta, potrebbe dover essere necessario inserire altre informazioni. Per esempio se si usa la tipologia "fixedValue", che fissa un valore specifico su tutta la patch a cui si riferisce, bisognerà ovviamente fornire un valore in forma scalare o vettoriale a seconda della grandezza.

Alla fine di questa sezione verranno riportati tutti i file ".c" contenenti le condizioni al contorno di tutte le grandezze per una maggiore comprensione e chiarezza.

Velocità "U"

Il flusso d'aria che investe il veicolo si muove ad una velocità di $50\frac{m}{s}$ in direzione "x", il terreno sottostante l'auto si muove alla medesima velocità nella stessa direzione e le ruote e il cerchione dell'auto ruotano in modo solidale con il movimento del terreno quindi, considerando il raggio della ruota, ad una velocità angolare di $\omega = 162.33\frac{rad}{s}$. Sulle pareti laterali e superiore del dominio sono state impostate le condizioni di free slip, così come è stato fatto sulla patch di raccordo tra ruota e terreno in quanto non costituisce un vero e proprio componente dell'auto.

Su tutte le patch che compongono l'ala la velocità è stata imposta pari a 0, imponendo quindi la condizione di "no slip" sulle pareti. All'outlet (sezione di uscita del dominio) è stata invece imposta una condizione "inletOutlet" in quanto dai primi calcoli è emerso un problema di ricircolo del flusso all'uscita, attraverso questa condizione è possibile impostare la velocità di backflow (flusso che rientra nel dominio di calcolo attraverso l'outlet) pari a 0.

Pressione "p"

Trattandosi di un caso che studia un flusso di fluido incomprimibile la pressione viene adimensionalizzata rispetto alla densità e la sua unità di misura risulta quindi $\frac{m^2}{s^2}$. Sulle pareti laterali, superiore e sulla patch di raccordo tra terreno e ruota, come nel caso della velocità, sono state imposte le condizioni di "free slip", all'ingresso è stato fissato il valore iniziale della pressione pari a 0 così come all'interno del dominio. Sulle pareti dove si forma lo strato limite è stata imposta la condizione "zeroGradient" essendo il gradiente di pressione all'interno dello strato limite in direzione perpendicolare alla parete pari a 0.

Energia cinetica turbolenta "k"

Il valore di "k" iniziale da inserire all'interno delle condizioni al contorno viene calcolato a partire da velocità, intensità turbolenta e scala di lunghezza turbolenta, questi dati sono stati decisi e forniti da Dallara e valgono rispettivamente $U = 50\frac{m}{s}$, $I_t = 0.1\%$ e $l_t = 0.1m$. Attraverso il calcolatore di Wolf Dynamics disponibile online [14] che si basa sull'equazione:

$$k = \frac{3}{2}(UI_t)^2 \tag{4.4}$$

è stato ricavato $k = 0.00375\frac{m^2}{s^2}$, questo è stato fissato come valore di partenza per il flusso indisturbato, mentre per il calcolo di questa grandezza sulle pareti della geometria, dove si ha la formazione di strato limite, è stata utilizzata come condizione al contorno "kqRWallFunction". Questa, si comporta da involucro rispetto alla condizione "zeroGradient" e ne eredita le caratteristiche. Sulle pareti laterali e superiore del dominio e sulla patch di raccordo tra suolo e ruota sono state imposte sempre le condizioni di "slip".

Tasso di dissipazione specifico dell'energia cinetica turbolenta "omega"

Il valore iniziale di "omega" (ω) è stato calcolato anch'esso attraverso il calcolatore di Wolf Dynamics attraverso la formula:

$$\omega = \frac{\sqrt{k}}{C_\mu l_t} \tag{4.5}$$

dove "k" è ottenuto con il procedimento sopra descritto e "C_μ" una costante legata ai modelli di turbolenza solitamente pari in questi casi a 0.09 [15]. Quindi è stato fissato per il flusso indisturbato e all'inlet il valore $\omega = 6.9[1/s]$ mentre sulle pareti dove si ha formazione di strato limite è stata utilizzata la condizione: "omegaWallFunction". Su pareti laterali, superiore e patch di contatto tra terreno e ruota anche in questo caso è stata impostata la condizione di "slip".

Viscosità cinematica turbolenta ”nut”

Le condizioni al contorno ”nut” (ν_t, viscosità cinematica turbolenta) e il suo valore iniziale sono state ricavate a partire dal valore di $\frac{\mu_t}{\mu}$, questo viene calcolato a partire dal valore di k e ω attraverso la formula:

$$\omega = \frac{\rho k}{\mu}\frac{\mu_t}{\mu} \tag{4.6}$$

Successivamente, sapendo che $\nu = \frac{\mu}{\rho}$, è possibile ricavare il valore di ν_t. Per il flusso indisturbato, ”internalField”, è quindi stato applicato $\nu_t = 0.0005435[m^2/s]$, mentre per le patch relative ad inlet e outlet è stata impostata la tipologia ”calculated” e come valore di partenza ν_t è stato fissato pari a 0. Per il calcolo di questa grandezza a parete è stata utilizzata la wall function: ”nutUSpaldingWallFunction” e sulle pareti laterali, superiore e sulla patch di contatto tra ruota e suolo sono state impostate le condizioni di ”slip”.

Nel caso in cui venisse utilizzato come modello di turbolenza Spalart-Allmaras non utilizzando questo alcun tipo di wall function il valore di ”ν_t” sulle pareti su cui si ha formazione di strato limite viene fissato pari a 0.

Viscosità turbolenta modificata ”nuTilda”

Questa grandezza viene inserita nelle condizioni al contorno solo nel caso in cui si dovesse utilizzare il modello di turbolenza Spalart-Allmaras. ”NuTilda” ($\tilde{\nu}$) rappresenta la viscosità turbolenta modificata, il suo valore iniziale utilizzato per il flusso indisturbato e fissato all'inlet viene calcolato moltiplicando la viscosità cinematica per un valore compreso tra 3 e 5 [16], nel nostro caso è stato quindi impostato un valore $\tilde{\nu} = 0.00007303\frac{m^2}{s}$. Il valore alla parete viene fissato pari a 0 e come per le grandezze precedentemente descritte sulle pareti superiore, laterali e sulla patch di contatto tra terreno e ruota sono state imposte le condizioni di ”slip”.

Codice

Come espresso all'inizio del capitolo, nelle pagine successive sarà possibile osservare la struttura di tutti i file sopra citati per una migliore comprensione di quanto detto e per visualizzare come sono state implementate tutte le informazioni in OpenFOAM.

```
FoamFile
{
    version 2.0;
    format binary;
    class volVectorField;
    location "";
    object U;
}
dimensions [0 1 -1 0 0 0 0];
internalField uniform (0 0 0);
boundaryField
{
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      slip;
        }

        p-outlet
        {
                type       inletOutlet;
                inletValue uniform (0 0 0);
                value      uniform (0 0 0);
        }

        ("v-inlet|w00-ground")
        {
                type      fixedValue;
                value     uniform (50 0 0);
        }

        ("wing-patches")
        {
                type      fixedValue;
                value     uniform (0 0 0);
        }

        ("w06b-fwlr-rim|w06b-fwlr-tyre-front|w06b-fwlr-tyre-rear")
        {
                type      rotatingWallVelocity;
                origin    (0.474991361 0.790915467 0.285761815);
                axis      (-0.000475119 -0.99893579 -0.046120081);
                omega     162.33;
        }
}
```

```
FoamFile
{
    version 2.0;
    format binary;
    class volScalarField;
    location "";
    object p;
}
dimensions [0 2 -2 0 0 0 0];
internalField uniform 0;
boundaryField
{
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      slip;
        }

        p-outlet
        {
                type      fixedValue;
                value     uniform 0;
        }

        ("v-inlet|w00-ground")
        {
                type      zeroGradient;
        }

        ("wing-patches")
        {
                type      zeroGradient;
        }

        ("w06b-fwlr-rim|w06b-fwlr-tyre-front|w06b-fwlr-tyre-rear")
        {
                type      zeroGradient;
        }
}
```

```
FoamFile
{
    version 2.0;
    format binary;
    class volScalarField;
    location "";
    object k;
}
dimensions [0 2 -2 0 0 0 0];
internalField uniform 0.00375;
boundaryField
{
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      slip;
        }

        p-outlet
        {
                type      zeroGradient;
        }

        v-inlet
        {
                type      fixedValue;
                value     uniform 0.00375;
        }

        ("wing-patches|w00-ground")
        {
                type      kqRWallFunction;
                value     uniform 0;
        }

        ("w06b-fwlr-rim|w06b-fwlr-tyre-front|w06b-fwlr-tyre-rear")
        {
                type      kqRWallFunction;
                value     uniform 0;
        }
}
```

```
FoamFile
{
    version 2.0;
    format binary;
    class volScalarField;
    location "";
    object omega;
}
dimensions [0 0 -1 0 0 0 0];
internalField uniform 6.9;
boundaryField
{
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      slip;
        }

        p-outlet
        {
                type      zeroGradient;
        }

        v-inlet
        {
                type      fixedValue;
                value     uniform 6.9;
        }

        ("wing-patches|w00-ground")
        {
                type      omegaWallFunction;
                value     uniform 0;
        }

        ("w06b-fwlr-rim|w06b-fwlr-tyre-front|w06b-fwlr-tyre-rear")
        {
                type      omegaWallFunction;
                value     uniform 0;
        }
}
```

```
FoamFile
{
    version 2.0;
    format binary;
    class volScalarField;
    location "";
    object nut;
}
dimensions [0 2 -1 0 0 0 0];
internalField uniform 5.435e-4;
boundaryField
{
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      slip;
        }

        p-outlet
        {
                type      calculated;
                value     uniform 0;
        }

        v-inlet
        {
                type      calculated;
                value     uniform 0;
        }

        ("wing-patches|w00-ground")
        {
                type      nutUSpaldingWallFunction; \\fixedValue (Spalart-Allmaras)
                value     uniform 0;
        }

        ("w06b-fwlr-rim|w06b-fwlr-tyre-front|w06b-fwlr-tyre-rear")
        {
                type      nutUSpaldingWallFunction; \\fixedValue (Spalart-Allmaras)
                value     uniform 0;
        }
}
```

```
FoamFile
{
    version 2.0;
    format binary;
    class volScalarField;
    location "";
    object nuTilda;
}
dimensions [0 2 -1 0 0 0 0];
internalField uniform 7.3036735e-05;
boundaryField
{
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      slip;
        }

        p-outlet
        {
                type      zeroGradient;
        }

        v-inlet
        {
                type      fixedValue;
                value     uniform 7.3036735e-05;
        }

        ("wing-patches|w00-ground")
        {
                type      fixedValue;
                value     uniform 0;
        }

        ("w06b-fwlr-rim|w06b-fwlr-tyre-front|w06b-fwlr-tyre-rear")
        {
                type      fixedValue;
                value     uniform 0;
        }
}
```

4.3.2 Directory ”constant”

All’interno della directory ”constant” sono contenuti principalmente 2 file: ”transportProperties” e ”turbulenceProperties” e una directory ”polyMesh”. All’interno di quest’ultima è possibile trovare tutti i file contenenti le informazioni relative alla griglia di calcolo.

Informazioni legate alla griglia di calcolo ”polyMesh”

All’interno di questa cartella il file sicuramente più importante è quello denominato ”boundary”. In questo file viene infatti impostata la tipologia associata ad ogni singola porzione del dominio. Per il corretto funzionamento di OpenFOAM è necessario che ad ogni patch all’interno di questo file venga associata la tipologia corretta per essere coerente con quanto inserito all’interno dei file nella directory ”0”. Dopo le informazioni riguardanti il file, già illustrate al sottocapitolo precedente, questa volta abbiamo un numero che simboleggia il numero di patch presenti nella griglia di calcolo, in questo caso 16.

```
FoamFile
{
    version 2.0;
    format binary;
    class polyBoundaryMesh;
    location "";
    object boundary;
}

16
(
        ("s-domain-limit-lat|s-domain-limit-sup|w00np-fwl")
        {
                type      patch;
        }

        ("p-outlet|v-inlet|wing-patches|wheel-patches|w00-ground")
        {
                type      wall;
        }
)
```

Parlando del caso in esame all’interno del seguente elaborato di tesi la tipologia base ”patch” può essere utilizzata per una qualsiasi condizione al contorno di tipo numerico o derivato disponibili in OpenFOAM, fatta eccezione per il caso in cui su una parte del dominio vengano impostate la tipologia ”fixedValue” nelle condizioni al contorno della velocità e ”zeroGradient” nelle condizioni al contorno della pressione, in quel caso la tipologia da inserire nel file ”boundary” sarà ”wall”.

Proprietà del fluido utilizzato ”transportProperties”

All’interno di questo file vengono elencate le proprietà del fluido utilizzato all’interno della simulazione e il modello di trasporto utilizzato. Nel nostro caso viene specificato un singolo valore di ”ν” quindi sotto ”transportModel” sarà necessario inserire ”newtonian”.

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      transportProperties;
}
transportModel  Newtonian;
nu              1.5e-05;
```

Proprietà del modello di turbolenza ”turbulenceProperties”

Infine, all’interno di ”turbulenceProperties” vengono inserite le informazioni a proposito del modello di turbolenza utilizzato, nel caso in esame sono stati utilizzati due modelli di turbolenza: un modello ad una equazione, Spalart-Allmaras, e uno a due equazioni, $k - \omega$ SST.

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      turbulenceProperties;
}
simulationType      RAS;
RAS
{
    RASModel        kOmegaSST;
    turbulence      on;
    printCoeffs     on;
}
```

4.3.3 Directory ”system”

All’interno della directory ”system” sono contenute principalmente le informazioni che controllano il processo di calcolo, le informazioni a proposito degli schemi di discretizzazione e dei solutori utilizzati per le varie grandezze. Possono inoltre essere presenti file riguardanti il controllo di altri processi quali: generazione e controllo della qualità delle griglie di calcolo, applicazioni di condizioni specifiche, impostazioni per il calcolo in parallelo sui vari core della CPU e molto altro. Di seguito verranno analizzati più da vicino i file principali utilizzati per lo svolgimento del seguente lavoro di tesi.

Controllo del processo di calcolo ”controlDict”

All’interno del file ”controlDict” sono contenute tutte le informazione generali relative a come viene eseguito il calcolo. Essendo considerato il caso analizzato stazionario il solutore utilizzato è stato ”simpleFoam”. Vengono poi inseriti l’istante di tempo iniziale del calcolo, ”startTime”, e quello finale, ”endTime”. Il ”timeStep” utilizzato è stato impostato pari a 1 poiché in un caso stazionario non vi è necessità di calcolare e controllare il numero di Courant. Per evitare spiacevoli inconvenienti relativi ad improvvise e inattese interruzioni del calcolo, che possono capitare, i risultati del calcolo sono stati salvati ogni 50 iterazioni utilizzando il comando ”writeTime” e andando ogni volta a sovrascrivere il risultato precedente inserendo ”purgeWrite 1;”.

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      controlDict;
}
application     simpleFoam;
startFrom       startTime;
startTime       0;
stopAt          endTime;
endTime         10000;
deltaT          1;
writeControl    timeStep;
writeInterval   1;
purgeWrite      0;
writeFormat     binary;
writePrecision  6;
writeCompression off;
timeFormat      general;
timePrecision   6;
runTimeModifiable true;
```

All’interno del file è poi inoltre possibile inserire varie funzioni con l’obiettivo di calcolare grandezze di interesse o svolgere azioni per modificare il calcolo durante il suo stesso avanzamento. Nel nostro caso in particolare sono stati monitorati tutti i coefficienti aerodinamici dell’ala completa e di ”main flap”, ”lower flap” e ”upper flap” singolarmente, per monitorarne l’andamento e verificare quando venisse effettivamente raggiunta la convergenza è tendenzialmente meglio basarsi su grandezze di interesse, piuttosto che sui residui del calcolo. Sono stati inoltre calcolati il valore di ”$y+$”, del coefficiente di pressione statico ”C_p” e del coefficiente di pressione totale ”C_{pT}” per visualizzare successivamente i risultati ottenuti. A seguire viene riportata come esempio la funzione utilizzata per calcolare i coefficienti aerodinamici dell’ala completa.

```
functions
{
    forceCoeffs_object_wing
    {
        type forceCoeffs;
        libs ("libforces.so");
        patches ("wing-patches");
        p p;
        U U;
        rho rhoInf;
        rhoInf 1.225;
        log true;
        CofR (0.241225 0.805970 0.286527);
        liftDir (0 0 1);
        dragDir (1 0 0);
        pitchAxis (0 1 0);
        magUInf 50;
        lRef 1.0;
        Aref 1;
        writeControl   timeStep;
        writeInterval  1;
    }
}
```

Configurazione degli schemi numerici ”fvSchemes”

Questo file fornisce le informazioni necessarie ad OpenFOAM per la discretizzazione nello spazio e nel tempo dei termini delle equazioni che governano il caso in esame, in particolare dei termini diffusivo, convettivo, gradiente e sorgente. Il setup può essere improntato più verso la stabilità e la convergenza del calcolo, a scapito della precisione, o viceversa. Questo file contiene una serie di sottogruppi che nello specifico sono:

- ddtSchemes: specifica come viene discretizzato il termine legato al tempo, in un caso stazionario come il nostro questo termine non viene risolto.
- gradSchemes: gestisce gli schemi numerici per i termini legati ai gradienti.
- divSchemes: gestisce gli schemi numerici per termini legati alla divergenza.
- laplacianSchemes: controlla la discretizzazione dei termini legati al laplaciano.
- interpolationSchemes: viene definito lo schema numerico con cui i valori presenti al centro delle celle viene interpolato sul centro delle facce.
- snGradSchemes: si occupa dell’interpolazione dei gradienti in direzione normale sulle facce.

Tra questi ”divSchemes” è forse quello più importante, al fine di non avere troppa diffusione sulla velocità è stato scelto di utilizzare come schema numerico per tutte le grandezze ”bounded Gauss linearUpwind”, la ”V” aggiuntiva nel caso della velocità sta ad indicare che il campo a cui si riferisce è un campo vettoriale. Nella pagina successiva è possibile osservare la struttura del file ”fvSchemes” utilizzata per i nostri calcoli.

Configurazione dei solutori di equazioni e algoritmi ”fvSolution”

All’interno del file ”fvSolution” sono contenute le informazioni per il controllo dei metodi di risoluzione dei sistemi lineari di equazioni algebriche discretizzate. Si osserva poi la presenza di un dictionary di controllo per l’algoritmo SIMPLE, questo controlla il processo iterativo che risolve le equazioni di velocità e pressione. Nel nostro caso sono stati impostati solo due correttori della non ortogonalità della mesh in quanto è stata ottenuta una mesh con una buona ortogonalità generale e inserirne di ulteriori avrebbe voluto dire appesantire inutilmente il calcolo e quindi rallentarlo. Sono inoltre presenti all’interno del file le impostazioni relative agli URF (Under Relaxation Factor) o fattori di rilassamento che aiutano a stabilizzare i risultati del calcolo incrementando però durata di questo. Per i nostri calcoli sono stati utilizzati tre differenti file ”fvSolution” contenenti tre step di URF crescenti in modo da far stabilizzare il calcolo nelle prime iterazioni e poi andare a ricercare una maggiore precisione dei risultati innalzando gli URF. Questo può essere fatto con una funzione apposita inserita nel dictionary ”function” nel file ”controlDict”.

Nelle pagine seguenti è possibile osservare la struttura dei file ”fvSchemes” e ”fvSolution”.

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
ddtSchemes
{
    default         steadyState;
}
gradSchemes
{
    default         cellLimited Gauss linear 1;
}
divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss linearUpwindV grad(U);
    div(phi,k)      bounded Gauss linearUpwind grad(k);
    div(phi,omega)  bounded Gauss linearUpwind grad(omega);
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default         Gauss linear limited 1;
}
interpolationSchemes
{
    default         linear;
}
snGradSchemes
{
    default         limited 1;
}
wallDist
{
    method          meshWave;
}
```

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
solvers
{
    p
    {
        solver           GAMG;
        tolerance        1e-6;
        relTol           0.01;
        smoother         DICGaussSeidel;
        minIter          1;
    }
    Phi
    {
        $p;
    }
    ("U|k|omega")
    {
        solver          PBiCGStab;
        preconditioner  DILU;
        tolerance       1e-8;
        relTol          0.01;
        minIter         1;
    }
}
SIMPLE
{
    nNonOrthogonalCorrectors 2;
    consistent yes;
}
relaxationFactors
{
    equations{U 0.7; k 0.7; omega 0.7;}
    fields{p 0.9;}
}
cache
{
    grad(U);
}
```

Capitolo 5

Risultati del calcolo diretto

All'interno di questo capitolo verranno analizzati i risultati ottenuti dal calcolo diretto. Verranno commentati i risultati ottenuti con le prime griglie di calcolo, poi successivamente migliorate, e osservato il confronto tra i risultati ottenuti con il noto software industriale Ansys Fluent, solitamente utilizzato da Dallara, e OpenFOAM. Verranno inoltre mostrati dei confronti tra i risultati ottenuti utilizzando il modello di turbolenza $k - \omega$ SST e Spalart-Allmaras e fatte le dovute considerazioni riguardo i successivi step legati all'utilizzo del metodo dell'aggiunto.

5.1 Risultati iniziali e modifiche alle griglie di calcolo

Nelle prime fasi di calcolo sono state utilizzate due mesh, sempre caratterizzate una da un valore di y+ elevato e l'altra da un valore di y+ ridotto. Portare i calcoli a convergenza è stato particolarmente complicato per motivi legati a celle non ideali ("bad cells") in diverse zone all'interno della griglia. Queste, sono celle che non garantiscono certi criteri di qualità sotto il punto di vista geometrico ad esempio possono avere una bassa ortogonalità oppure una elevata "skewness" o "aspect ratio", tutti questi fattori potrebbero essere causa di errori o nella peggiore delle ipotesi potrebbero portare alla divergenza del calcolo. Tendenzialmente quando si generano griglie di calcolo strutturate si cerca di ottenere come celle degli esaedri perfetti, cioè dei cubi, questo non è possibile nel caso in cui vengano analizzate geometrie complesse come quella del caso in esame. Tuttavia, per la corretta riuscita del calcolo, è fondamentale che la mesh abbia la massima qualità possibile sotto il punto di vista geometrico. Per ottenere una buona convergenza è stato utilizzato uno schema di discretizzazione del tipo "Gauss upwind" ma i risultati prodotti con OpenFOAM non rispettavano quelli ottenuti con Ansys Fluent a causa della eccessiva diffusione sulla velocità data dall'utilizzo di questo metodo. Si è pensato quindi di andare a migliorare le griglie in modo da ottenere una maggiore stabilità durante il calcolo, anche utilizzando schemi di discretizzazione caratterizzati da una maggiore correttezza del risultato, e una più elevata precisione dei risultati finali.

Sono state quindi generate due nuove mesh contenenti diversi miglioramenti: la strategia utilizzata per generare la mesh è stata di tipo ”snap”, la fase di ”snapping” consiste in un processo iterativo in cui, a partire dalla ”castellated mesh”, la griglia viene modificata e proiettata sui confini della geometria attorno alla quale deve essere generata. La ”castellated mesh”, a sua volta, si ottiene a partire dalla griglia strutturata di esaedri, eliminando le celle all'interno (o all'esterno a seconda delle applicazioni) della geometria che si vuole analizzare. In questo modo si ottiene una forma simile alla geometria finale ma non liscia, bensì formata dalle celle ancora di forma esaedrica che intersecano i confini della geometria.
Sono stati aggiunti dei raccordi dove necessario nella geometria, soprattutto nelle zone in cui erano presenti spigoli vivi o troncature, sono stati poi aggiunte diverse ”box” di raffinamento sui vari flap e nella zona relativa al pilone dell'ala per evitare la presenza di ”bad cells” aumentando la risoluzione della griglia in quelle zone. Infine, è stato rivisto il boundary layer per assicurarsi che non fosse degenere in nessuna delle due mesh e sono stati effettuati ulteriori raffinamenti locali su bordo di attacco e di uscita dei vari flap e sugli spigoli delle ruote, soprattutto sul cerchio e nelle parti di collegamento tra cerchio e gomma.

Le due nuove griglie di calcolo ottenute contano rispettivamente per quella low y+ e quella high y+ un numero di elementi pari a 12 milioni e 16 milioni con un boundary layer coverage del 99.8%. La maggior precisione delle mesh, come già accennato, ha permesso l'utilizzo di fattori di rilassamento di valore maggiore, che garantiscono una maggiore precisione dei risultati.

Dopo una serie di calcoli iniziali con le nuove griglie sono emerse due classi di risultati: una con un C_l del ”main flap” nell'intorno di 0.55 e una in cui il C_l del ”main flap” si assesta nell'intorno di 0.35. Andando ad osservare la distribuzione del C_p sull'ala le differenze sono molto marcate e visibili come è possibile osservare nelle figure 5.1 e 5.2.

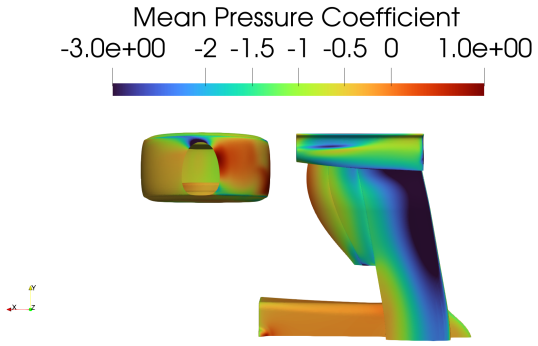


Figura 5.1: Soluzione ”Gauss linearUpwind” Coefficiente di ”lift”: $C_l = -0.85$

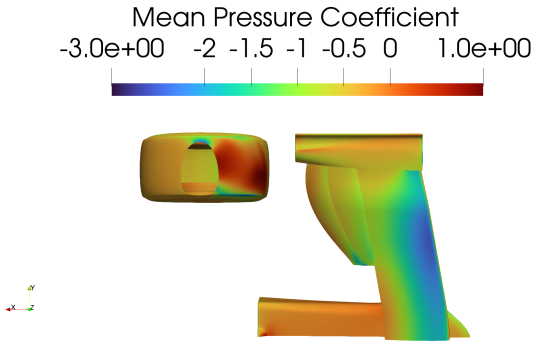


Figura 5.2: Soluzione ”Gauss upwind” Coefficiente di ”lift”: $C_l = -0.55$

L'ottenimento di uno o l'altro risultato dipende dallo schema di discretizzazione iniziale utilizzato all'interno di "fvSchemes" nel dictionary "divSchemes": se, per cercare di ottenere una migliore convergenza, si inizia il calcolo utilizzando uno schema del tipo "Gauss upwind" e dopo un certo numero di iterazioni si passa a "Gauss linearUpwind" si ottiene una soluzione con un C_p maggiore sull'estradosso dell'ala e conseguentemente meno "downforce" e un valore di C_l maggiore (Figura 5.2), soprattutto in corrispondenza della porzione di ala davanti alla ruota. Se invece si utilizza uno schema "linearUpwind" dall'inizio del calcolo si ottiene una soluzione con un C_p più negativo sull'estradosso e quella che a prima vista sembra una minore influenza della ruota sull'ala (Figura 5.1). Questo problema è probabilmente legato all'instazionarietà del caso, generata in particolar modo dalla presenza della ruota che, essendo un corpo tozzo, genera una forte scia dietro di se e molti vortici che aumentano, appunto, la non-stazionarietà del caso. Nel momento in cui si cerca quindi di risolvere il caso utilizzando un solutore stazionario ("simpleFoam" nel nostro caso) a seconda della condizione iniziale che si utilizza, a parità di tutte le altre condizioni (griglia di calcolo, condizioni al contorno, numero di iterazioni, ...), si ottiene una biforcazione dei risultati senza inizialmente sapere quale delle due casistiche sia quella corretta.

Cenni sul calcolo instazionario

Un modo per capire quale sia effettivamente la soluzione da tenere in considerazione è effettuare un calcolo instazionario. Questo tipo di processi richiedono molta potenza di calcolo per diversi motivi. Innanzitutto, nel caso "unsteady" è necessario risolvere i termini, non più pari a 0, delle equazioni governanti del flusso relativi alle variazioni delle grandezze nel tempo. In secondo luogo nel caso in cui vengano studiate griglie molto fitte, con celle molto piccole rispetto alle dimensioni caratteristiche del sistema, potrebbe essere necessario ridurre il "time step" a valori molto piccoli per mantenere il Numero di Courant (CFL) il più possibile vicino a 1. Questo perché è necessario, istante per istante, seguire la singola particella di fluido attraverso il suo movimento nel dominio di calcolo, per farlo correttamente è necessario che ad ogni "time step" la particella percorra una distanza pari al massimo a quella di un elemento della mesh. Quindi, per far si che ogni particella venga seguita correttamente lungo tutto il suo percorso all'interno del dominio potrebbero essere necessari un numero molto grande di "time step". Considerando poi che ad ogni istante di tempo vengono effettuate altre sotto iterazioni, in modo da raggiungere la convergenza numerica dello specifico istante, il calcolo potrebbe diventare molto costoso in termini di tempo per mesh fitte con molti elementi.

Per trovare una soluzione precisa ed esatta occorre quindi calcolare correttamente il "timeStep" da inserire nel file "controlDict", in questo caso non più pari a 1, in modo da ottenere un numero di Courant (CFL) il più possibile vicino al valore unitario. OpenFOAM permette di gestire e variare automaticamente il "timeStep" in questi casi basandosi ad ogni iterazione sul massimo numero di Courant ottenuto, definito come:

$$CFL = \frac{u\Delta t}{\Delta x} \tag{5.1}$$

Fornisce informazioni a proposito di come le informazioni attraversano la griglia di calcolo (cioè la mesh) in un determinato "timeStep". Prendendo come esempio una particella che si muove a velocità pari a u il numero di Courant (CFL) ci dice in un determinato intervallo Δt quanti elementi di mesh di dimensione Δx abbiamo attraversato.

E' logico pensare che per seguire correttamente una particella nel suo moto all'interno del dominio di calcolo si debba conoscere la posizione della particella in ogni elemento di mesh e per questo motivo se si vogliono ottenere dei buoni risultati è necessario rimanere con un $CFL \simeq 1$. Siccome però il nostro interesse è stato solo quello di ottenere una soluzione che ci fornisse alcune indicazioni riguardo ai risultati ottenuti con il calcolo stazionario, non è stato necessario rispettare una simile condizione e si è quindi utilizzato un "timeStep" più elevato per ridurre il tempo di calcolo. Inoltre, è stato effettuato anche un calcolo rimuovendo la ruota in modo da eliminare la principale causa di instazionarietà. Per ottenere un risultato totalmente esatto servirebbe una "DNS" (Direct Numerical Simulation), con la quale le equazioni di Navier-Stokes vengono risolte direttamente, senza l'utilizzo di modelli di turbolenza. Per una geometria così complicata e con i mezzi a nostra disposizione un calcolo di questo tipo non è stato preso in considerazione a causa del troppo elevato costo computazionale. Infatti, la DNS risolve tutte le scale turbolente da quelle macroscopiche fino alle scale di Kolmogorov, dove i vortici più piccoli vengono dissipati sotto forma di calore a causa della viscosità del flusso. Per farlo, è necessario utilizzare griglie di calcolo molto fitte, specialmente nel caso in cui venissero analizzate geometrie particolarmente complicate, questo ovviamente appesantisce il calcolo. Inoltre, trattandosi di un calcolo instazionario anche in questo caso vi sono tutti i problemi legati al "time step" di cui sopra. In aggiunta a questo, le equazioni di Navier-Stokes devono essere integrate e risolte nella loro forma non mediata e questo complica ulteriormente il procedimento.

I risultati ottenuti dai calcoli effettuati dopo aver rimosso la ruota e quelli ottenuti dal calcolo "unsteady" ci hanno permesso di considerare come corretta la soluzione caratterizzata da un C_p più negativo nella zona vicina all'"endplate" (Figura 5.1. Successivamente, per cercare di separare l'instazionarietà della ruota dall'ala, ed eliminare quindi la duplicità del risultato, abbiamo provato a spostare leggermente più indietro rispetto alla posizione originale la ruota seguendo vari step, alla fine la posizione migliore si è rivelata essere quella che prevedeva uno spostamento di 30 centimetri della ruota in direzione "x", cioè 30 centimetri più indietro rispetto all'ala. Per una migliore comprensione lo spostamento viene mostrato in figura 5.3 e figura 5.4.

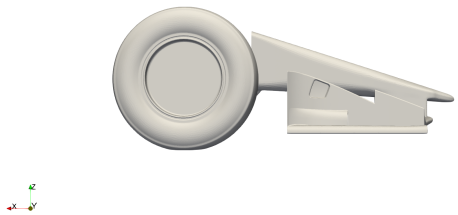


Figura 5.3: Ruota nella posizione originale

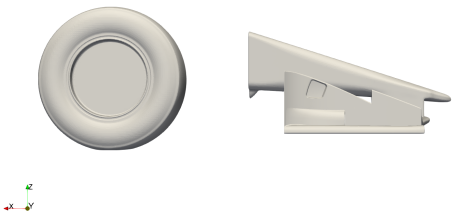


Figura 5.4: Ruota nella posizione modificata

E' stato possibile utilizzare questo procedimento principalmente per due motivi: il principale scopo della trattazione è l'ottimizzazione dell'ala anteriore, inoltre, la geometria che stiamo utilizzando è una geometria semplificata che non tiene conto di tutti gli elementi della sospensione, che tipicamente si trovano su un'auto a ruote scoperte, dei condotti per il raffreddamento dei freni e del resto della vettura che in questo caso non è presente. Questo, ci lascia margine di manovra per quanto riguarda lo spostamento della ruota, considerata solo come elemento di disturbo, rispetto all'ala.

Inoltre, gli spostamenti sono ovviamente piccoli e poco rilevanti per quanto riguarda globalmente il caso e il numero di elementi di mesh è abbastanza ridotto rispetto a casi in cui si studiano le geometrie complete o la macchina completa. Per concludere, quindi, questo tipo di azione è ritenuta accettabile in un lavoro di tesi come quello descritto dal seguente elaborato.

Una volta definita la geometria e generate le mesh finali e definitive sono stati eseguiti i calcoli per entrambe le griglie. Per prima cosa è stato fatto un confronto tra i risultati ottenuti con OpenFOAM e quelli ottenuti, utilizzando lo stesso setup e le medesime mesh, con Ansys Fluent.

5.2 Confronto tra OpenFOAM e Ansys Fluent

Per confrontare i risultati ottenuti con OpenFOAM e Ansys Fluent sono state considerate 6 sezioni di controllo sui flap dell'ala (Figura 5.5) disposte in modo da coprire nel modo più ampio e uniforme possibile tutti le zone di interesse di questa. In particolare i piani di sezione hanno tutti normale parallela all'asse y e da destra verso sinistra si troveranno rispettivamente a:

- $y/s = 0.37$
- $y/s = 0.42$
- $y/s = 0.53$
- $y/s = 0.63$
- $y/s = 0.74$
- $y/s = 0.79$

Dove "y" è chiaramente lo spostamento lungo y e "s" è il "semi-span" dell'ala, vale a dire la metà della larghezza.

Lungo tutti i profili dei flap dell'ala, sulle sezioni visibili in rosso in figura 5.5, è stato estratto il coefficiente di pressione "C_p" attraverso Paraview e sono stati poi generati dei grafici utilizzando uno script Python dedicato. Questo, permette all'utente di scegliere se visualizzare i grafici singolarmente, per una visione più dettagliata, o una sola immagine comprensiva di tutte le sezioni affiancate, per una visione d'insieme.

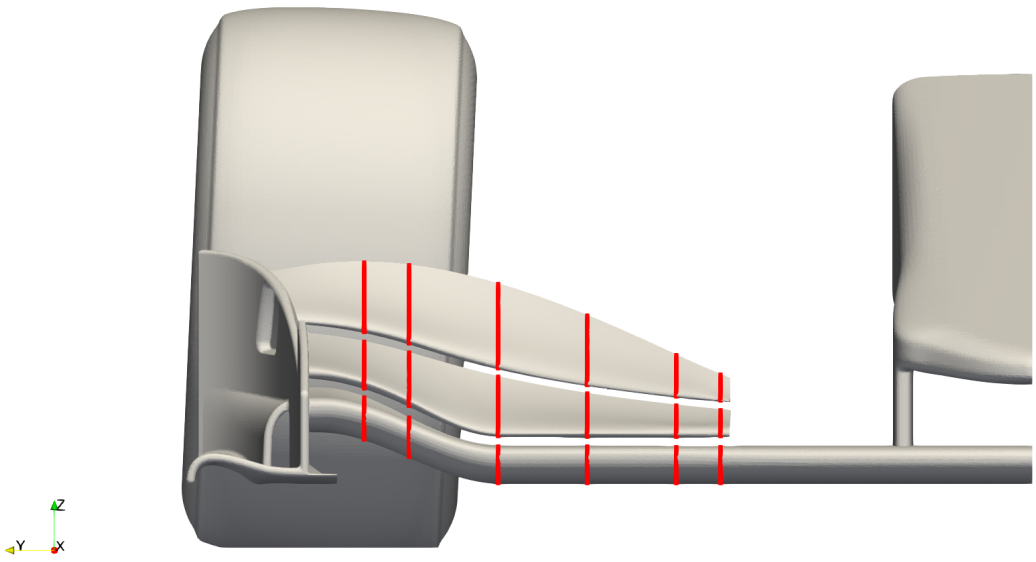


Figura 5.5: Sezioni di controllo sull'ala

5.2.1 Confronto dei risultati per la griglia di calcolo "hyp"

Nelle pagine successive, dalla figura 5.6 alla figura 5.11, è possibile vedere i confronti tra i risultati ottenuti con OpenFOAM e Ansys Fluent utilizzando la mesh caratterizzata da un valore di "$y+$" elevato (hyp). Normalmente i grafici del Coefficiente di Pressione (C_p) di un profilo vengono mostrati con l'asse delle ordinate invertito in modo che la parte alta della curva corrisponda all'estradosso del profilo alare, cioè quella che di solito è la parte alta del profilo, e la parte bassa all'intradosso. Il seguente elaborato esamina un'ala utilizzata in campo automobilistico, quindi in questo caso non è stato necessario invertire gli assi in quanto l'estradosso del profilo si trova già nella parte bassa. Nei grafici troveremo quindi sull'asse delle ascisse la coordinata "x" adimensionalizzata rispetto alla corda del "main flap" e sull'asse delle ordinate, non invertito, il Coefficiente di Pressione (C_p). Lo stesso discorso si applica a tutti gli altri grafici relativi a questo coefficiente durante la seguente trattazione.

In questo caso in particolare, si osserva come le curve relative ai due software siano abbastanza vicine tra di loro. Si nota, tuttavia, una leggera differenza tra le curve di tutti i flap in figura 5.6. Invece, in figura 5.7 sono molto più fedeli tra loro le curve relative al "lower flap", mentre sono ancora visibili delle piccole differenze su "main flap" e "lower flap".

Le sezioni analizzate in figura 5.8 e 5.9 mostrano invece un'ottima somiglianza tra i risultati di OpenFOAM e Ansys Fluent su tutte le curve fatta eccezione per uno leggero scostamento nella parte vicina al bordo di attacco sull'estradosso del "main flap".

Infine, sulle ultime due sezioni in figura 5.10 e figura 5.11 si osservano solo alcune piccole differenze tra le curve di "lower flap" e "upper flap".

Nelle pagine successive è possibile visualizzare tutti i grafici appena descritti.

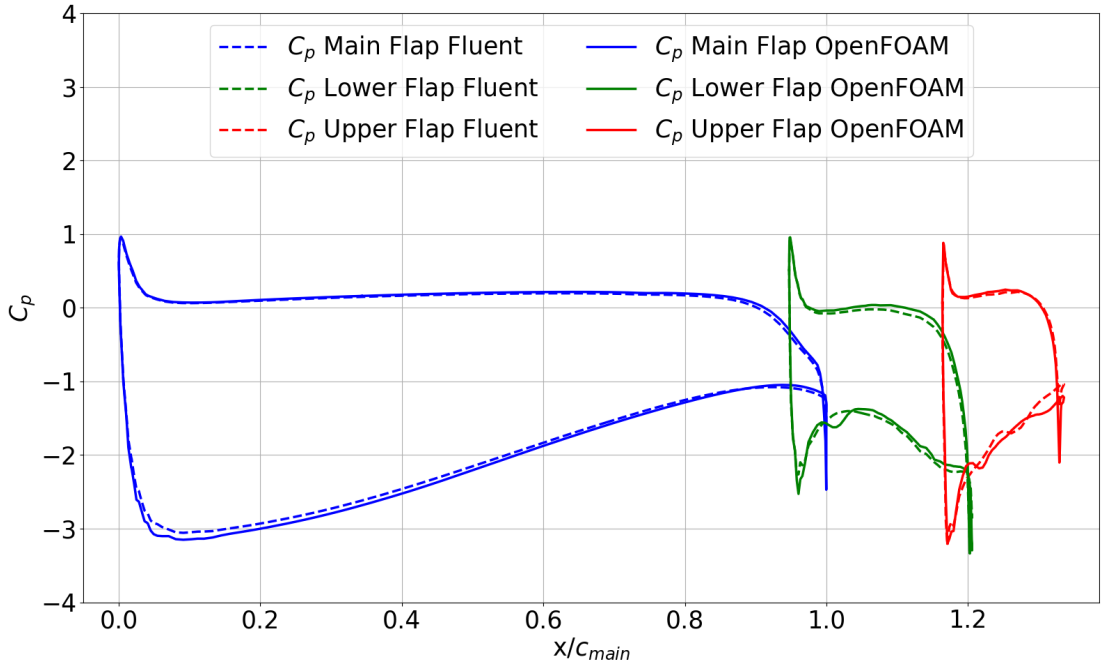


Figura 5.6: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.37$

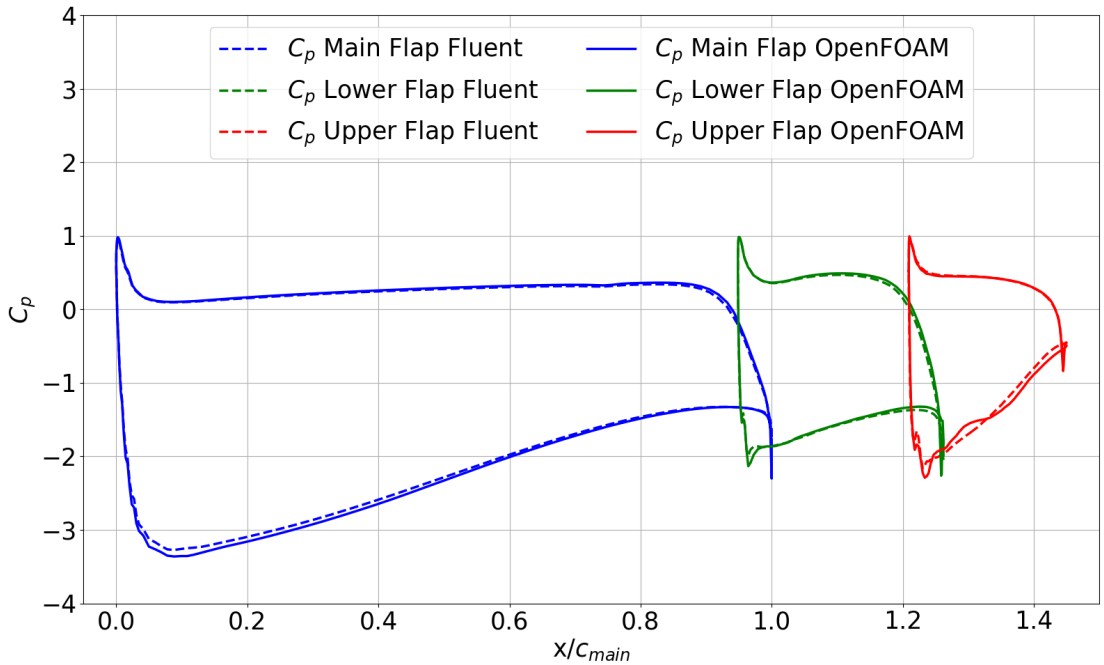


Figura 5.7: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.42$

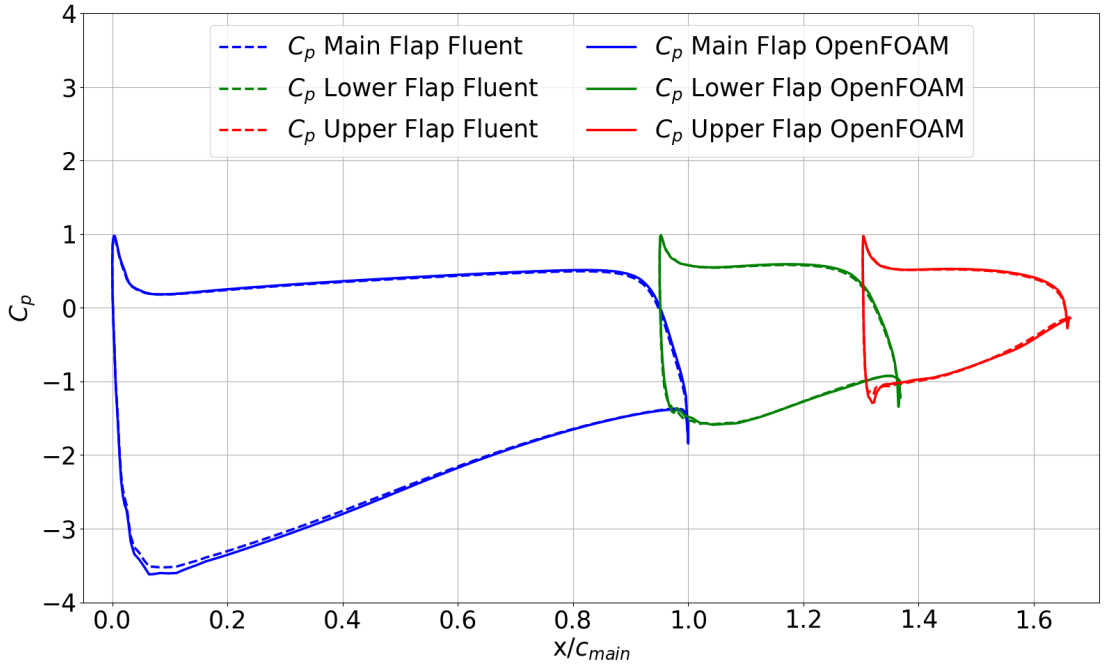


Figura 5.8: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.53$

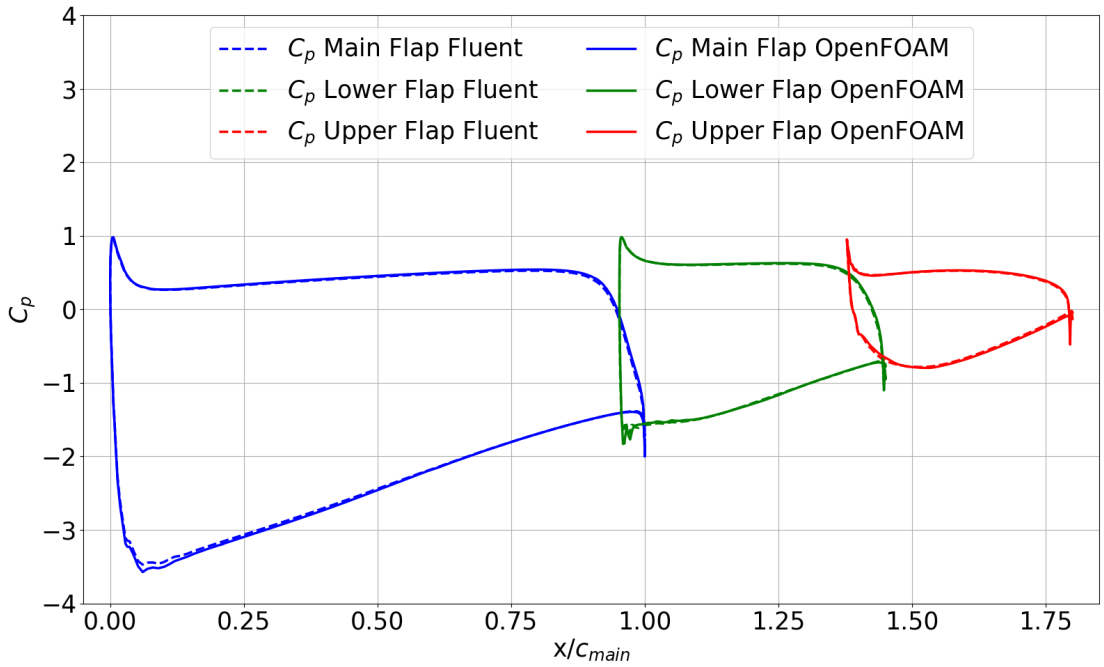


Figura 5.9: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.63$

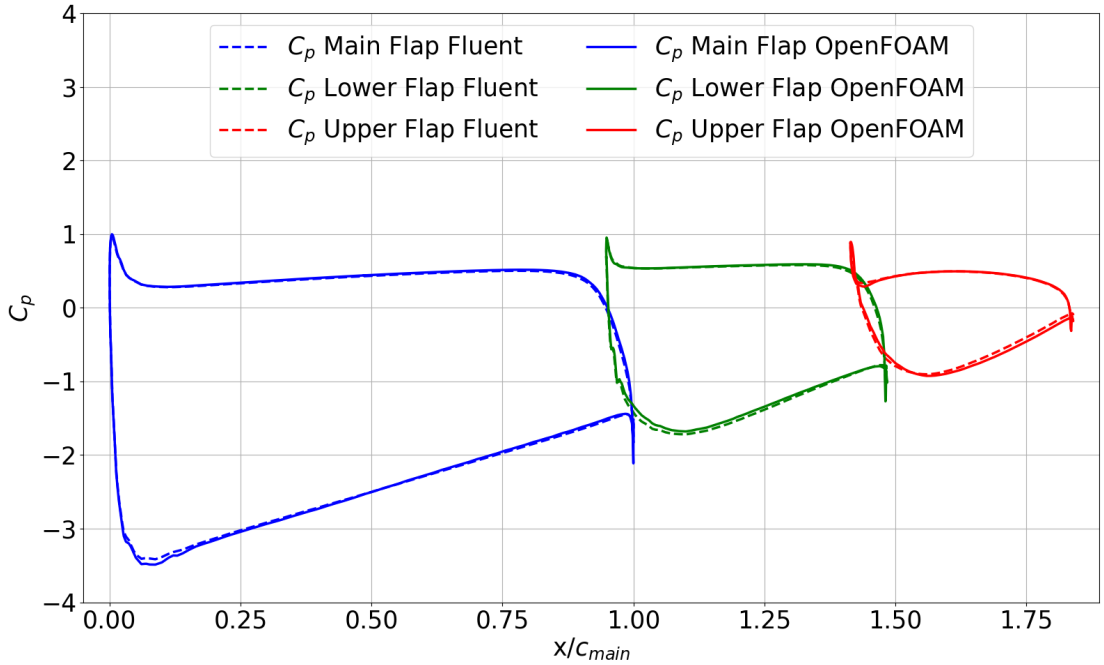


Figura 5.10: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.74$

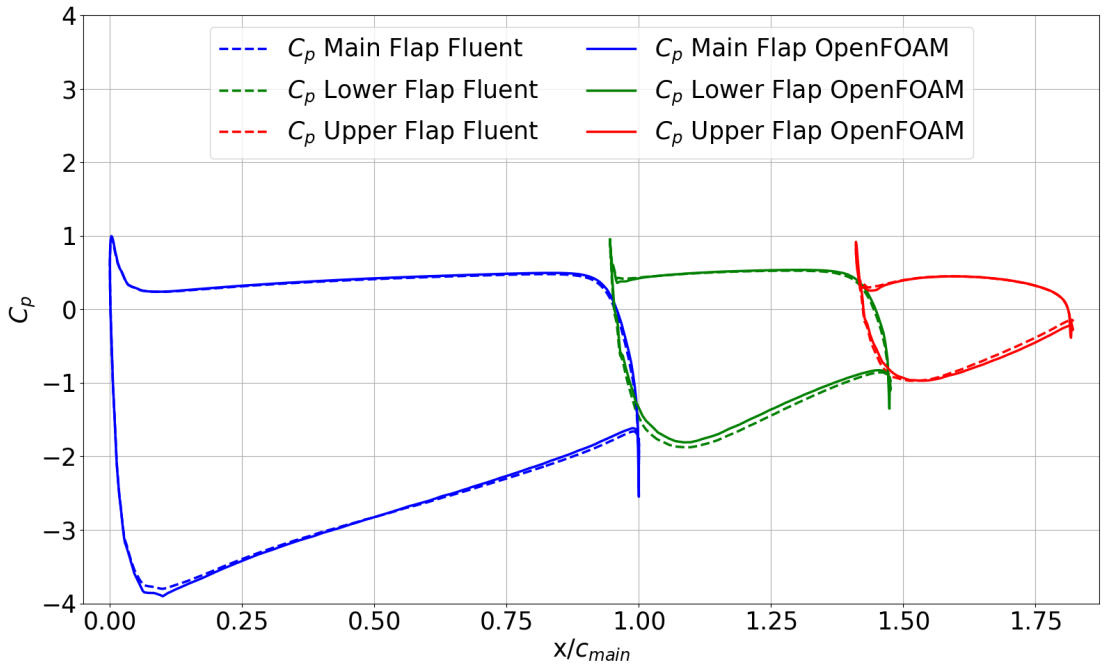


Figura 5.11: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh hyp - $y/s = 0.79$

5.2.2 Confronto dei risultati per la griglia di calcolo ”lyp”

Nel seguente sottocapitolo è possibile osservare il confronto tra i risultati ottenuti con OpenFOAM e Ansys Fluent utilizzando la mesh lyp, vale a dire caratterizzata da un valore di ”$y+$” ridotto, dalla figura 5.12 fino alla figura 5.17. Esattamente come nel caso precedente anche questa volta troviamo il Coefficiente di Pressione (C_p) sull’asse delle ordinate e la coordinata ”x” adimensionalizzata rispetto alla corda del ”main flap” sulle ascisse.

Nelle prime 4 sezioni analizzate, quindi in figura 5.12, figura 5.13, figura 5.14 e figura 5.15 le curva ottenute con OpenFOAM sono molto vicine, quasi completamente sovrapposte, a quelle ottenute con Ansys Fluent. Gli unici aspetti da segnalare si trovano sulle due sezioni rimanenti.

In figura 5.16 e figura 5.17, infatti, si nota un leggero discostamento delle curve di OpenFOAM da quelle di Ansys Fluent. Sembra che sull’estradosso del ”lower flap” OpenFOAM sottostimi la leggera separazione dello strato limite, legata probabilmente alla forte incidenza che caratterizza ”lower flap” e ”upper flap” e al gradiente di pressione avverso, aiutato dall’aumento di pressione in quella zona dato dalla presenza della ruota. Questo, provoca un aumento del ”C_p” verso il bordo di uscita dei due flap che è più evidente nei risultati di Ansys Fluent rispetto a quelli di OpenFOAM. Essendo lo strato limite meno separato sul ”lower flap” ne consegue poi una maggiore deportanza data dal ”upper flap” che verrà investito da un flusso più pulito.

Quanto appena descritto è visibile nelle figure alle pagine successive.

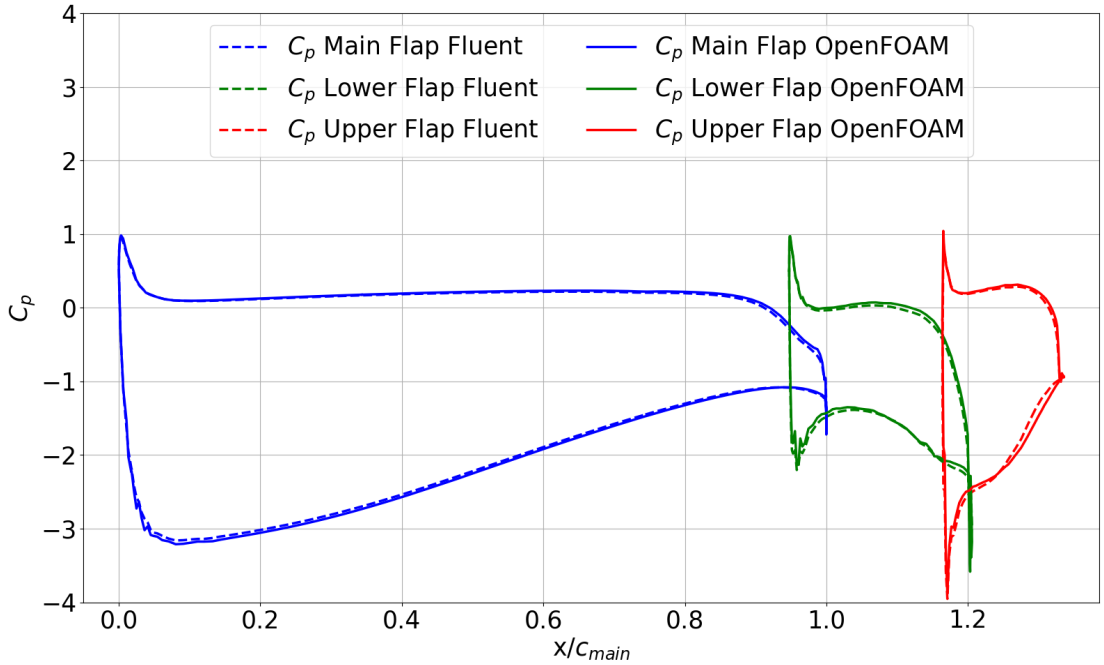


Figura 5.12: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.37$

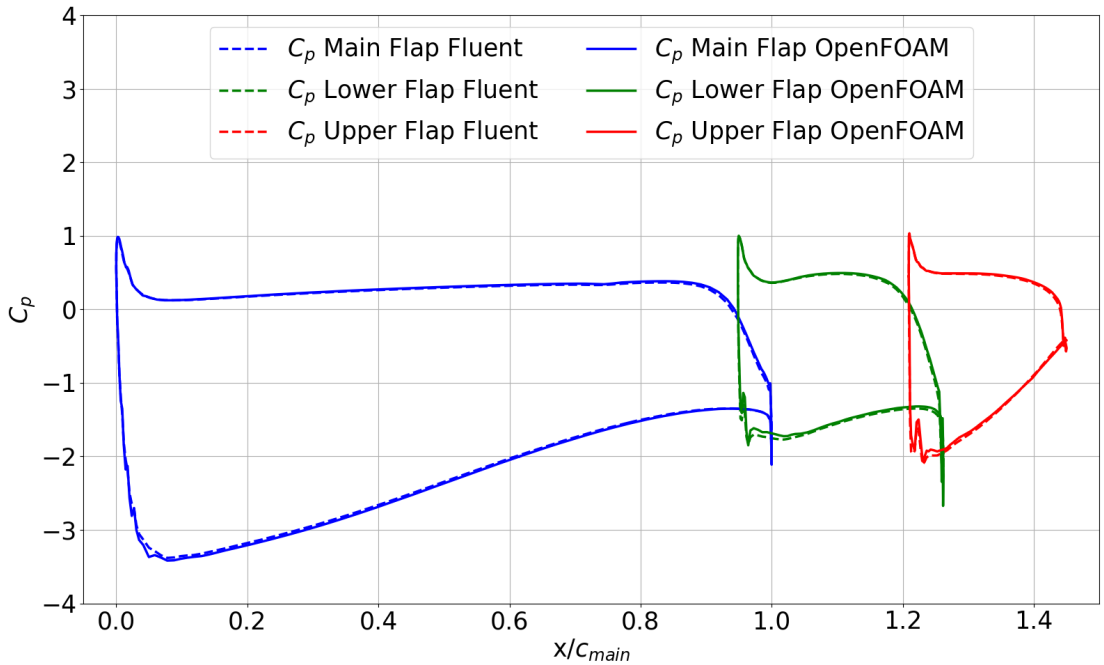


Figura 5.13: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.42$

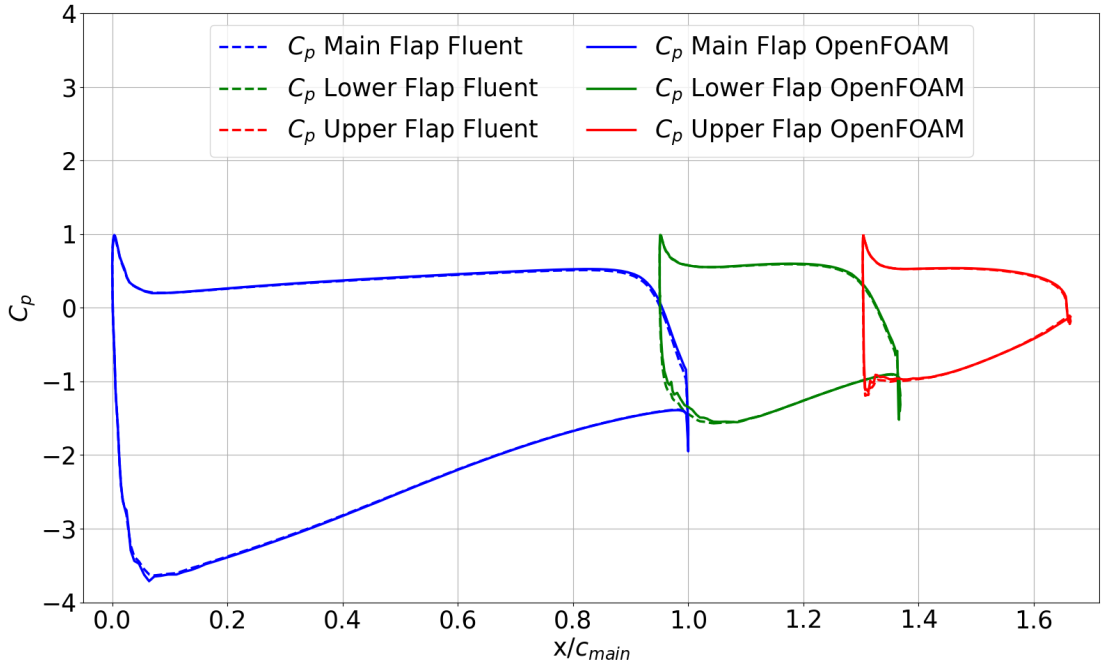


Figura 5.14: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.53$

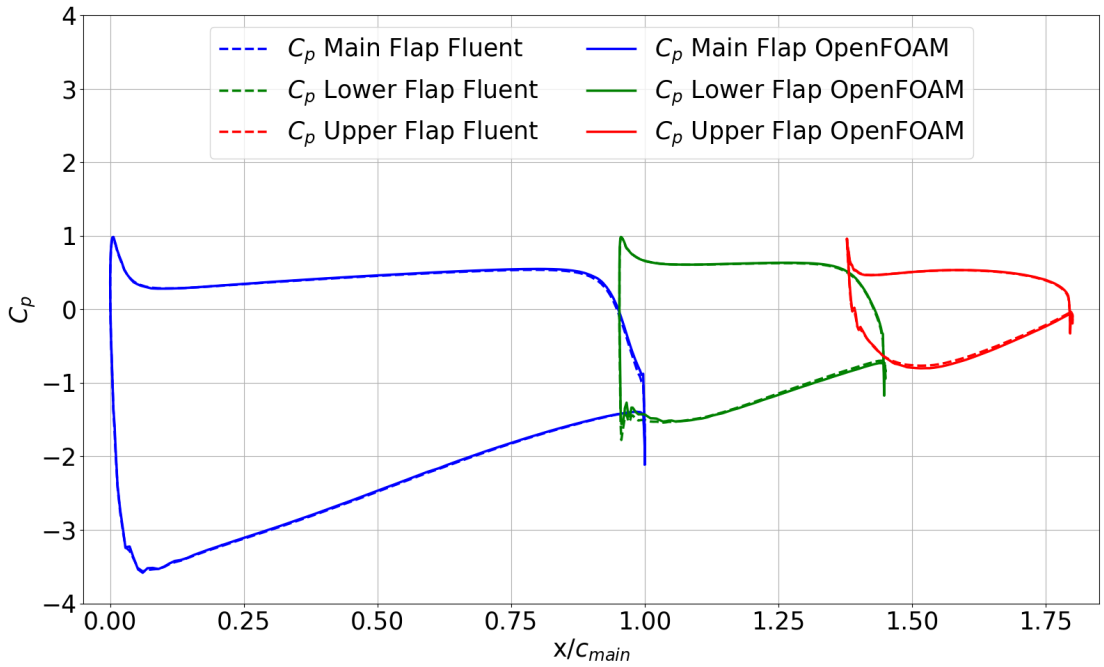


Figura 5.15: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.63$

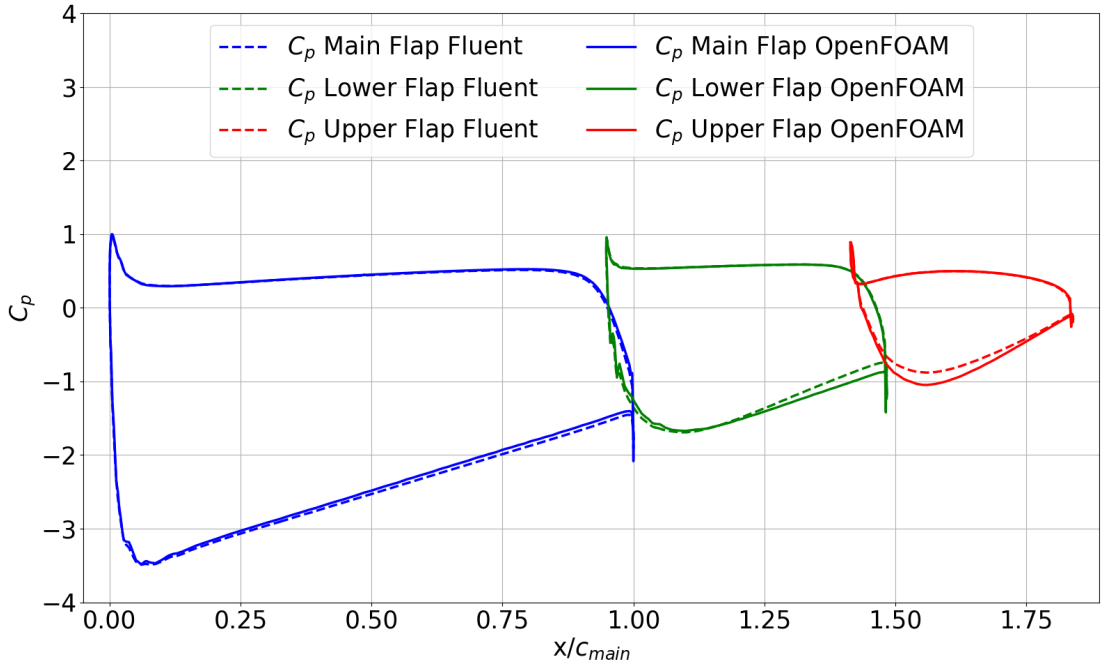


Figura 5.16: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.74$

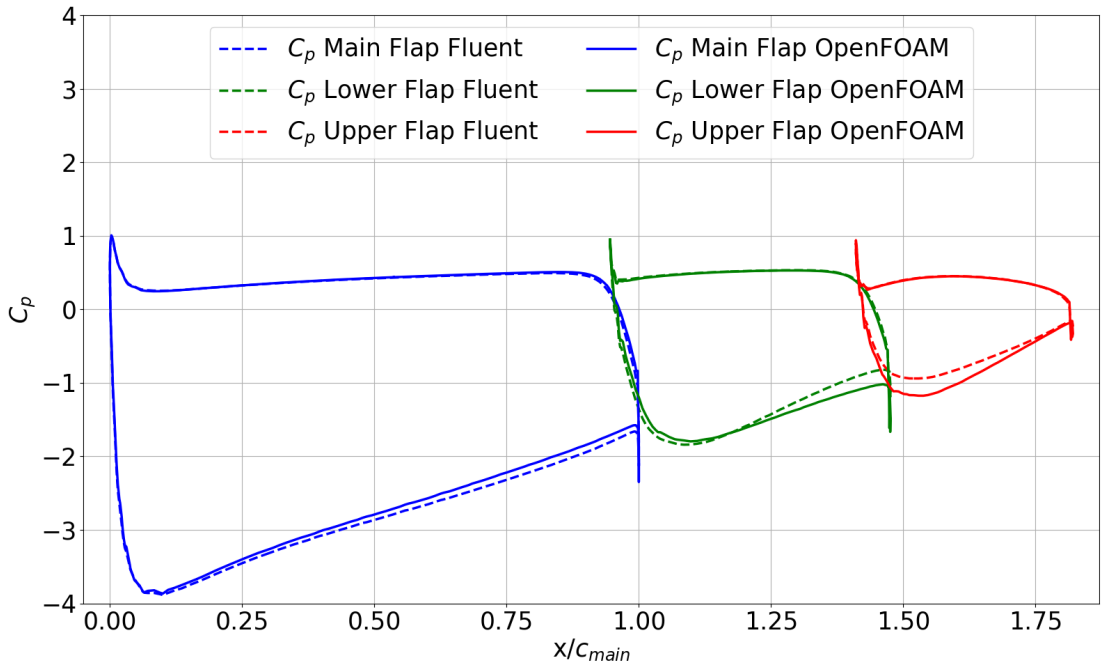


Figura 5.17: Confronto del C_p tra OpenFOAM e Ansys Fluent - Mesh lyp - $y/s = 0.79$

5.3 Risultati Spalart-Allmaras

Sono stati analizzati anche i risultati del calcolo diretto ottenuti utilizzando il modello di turbolenza Spalart-Allmaras, confrontandoli con quelli ottenuti con il modello $k-\omega$ SST. Questo perché è stato necessario durante la fase di ottimizzazione avere dei risultati più stabili e con residui del calcolo più bassi rispetto a quelli offerti dalla soluzione ricavata con $k-\omega$ SST, questo aspetto verrà visto in modo più dettagliato nel Capitolo 7. Tuttavia, il modello Spalart-Allmaras non è il modello ritenuto più adatto per lo studio di una geometria particolarmente complessa come quella in esame, è stato quindi necessario fare prima di tutto un confronto per verificare che i risultati ottenuti non fossero troppo lontani da quelli considerati corretti del modello $k-\omega$ SST.

5.3.1 Distribuzione di "y+" e del Coefficiente di Pressione (C_p)

Anche in questo caso per prima cosa è stata confrontata la distribuzione del "C_p" sulle stesse sezioni di controllo viste precedentemente, confrontando questa volta solo i risultati ottenuti dalla griglia di calcolo caratterizzata da un valore di "$y+$" ridotto. Questo, perché al contrario di $k-\omega$ SST il modello di turbolenza Spalart-Allmaras non è "$y+$ insensitive" e produce risultati migliori nel momento in cui si utilizzano mesh con un valore di "$y+$" compreso tra 1 e 5. In figura 5.18 e figura 5.19 è possibile osservare i contour di "$y+$" sull'ala per i due modelli di turbolenza utilizzati. Si osserva come le due figure siano diverse tra di loro soprattutto nella zona relativa alla ruota, comunque, in entrambi i casi, la quasi interezza della geometria è caratterizzata come detto precedentemente da un valore di "$y+$" compreso tra 1 e 5.

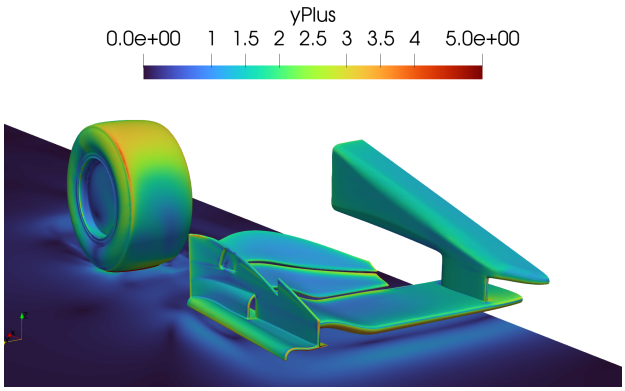


Figura 5.18: Contour y+ $k-\omega$ SST

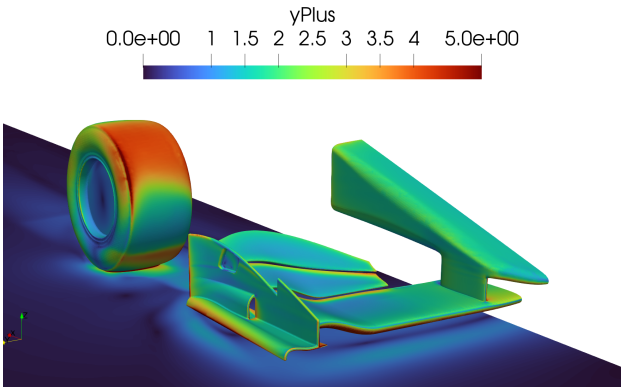


Figura 5.19: Contour y+ Spalart-Allmaras

Passando ora a parlare della distribuzione di "C_p" nelle pagine successive dalla figura 5.20 alla figura 5.25 è possibile notare come sull'estradosso dei profili il modello Spalart-Allmaras restituisca un valore di "C_p" decisamente inferiore rispetto a quello dato dal modello $k-\omega$ SST. Questo avviene in modo abbastanza uniforme su tutti i flap e su tutte le sezioni analizzate.

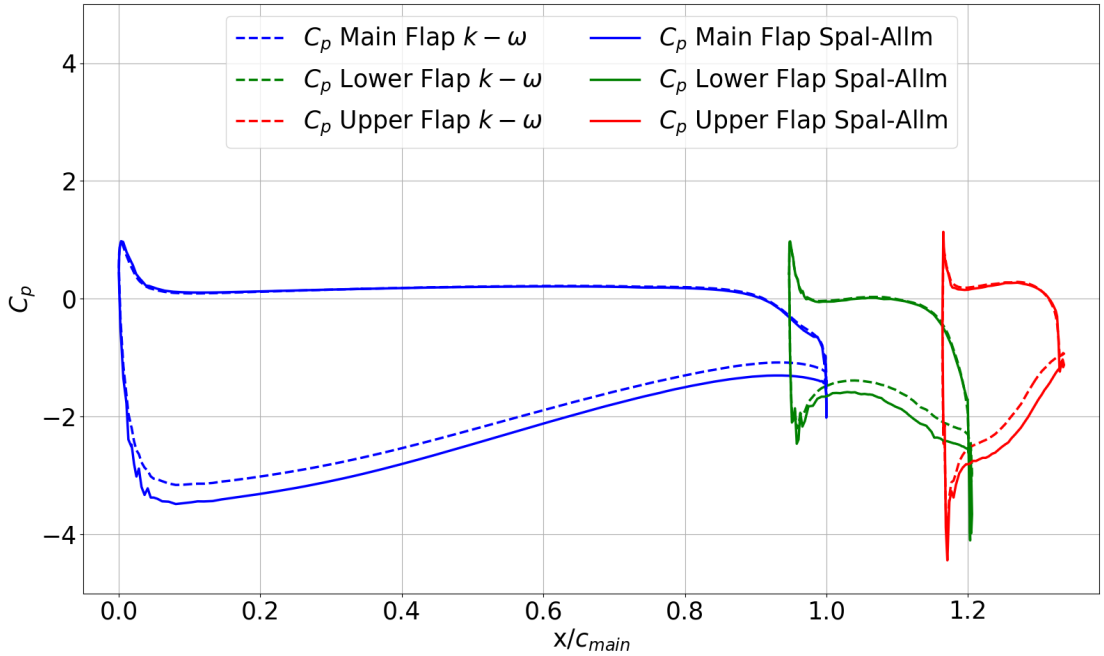


Figura 5.20: Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.37$

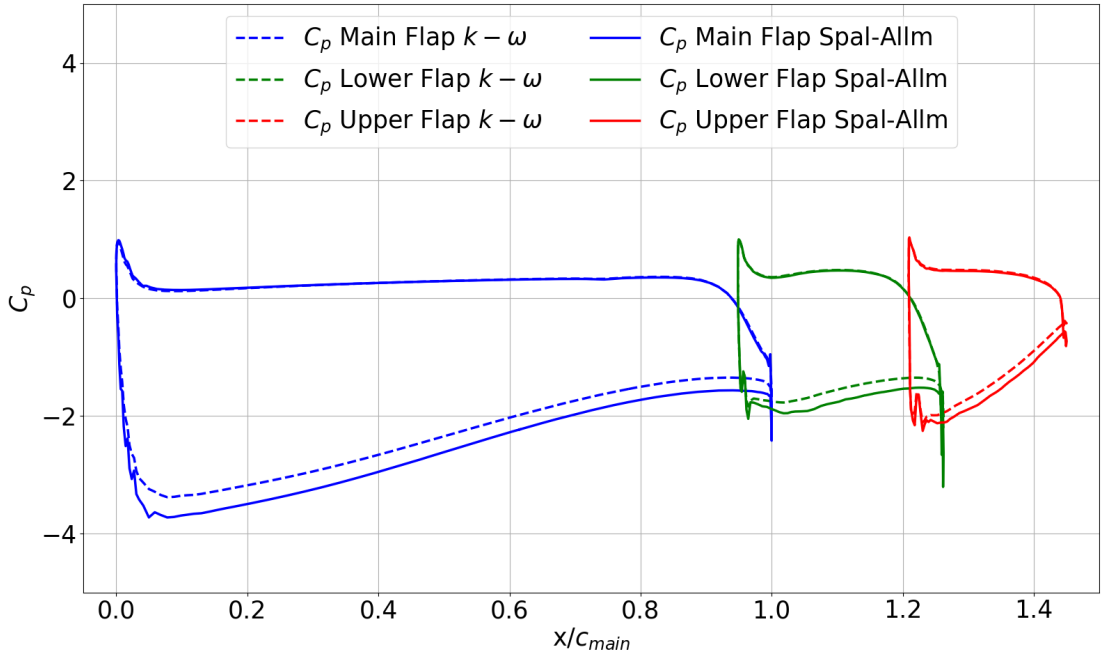


Figura 5.21: Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.42$

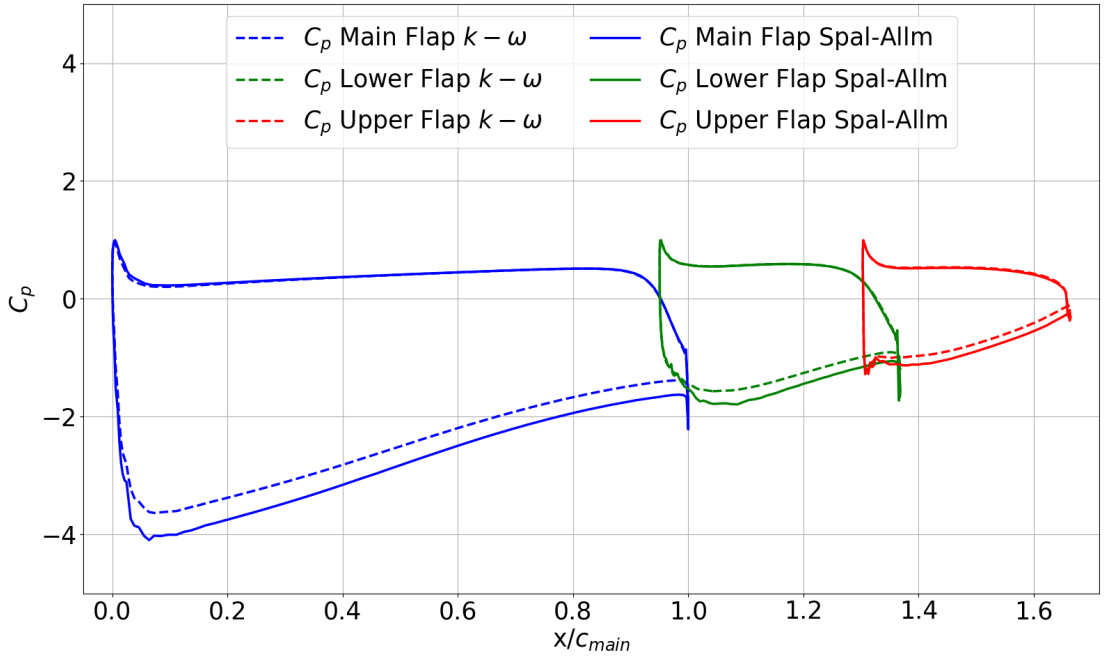


Figura 5.22: Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.53$

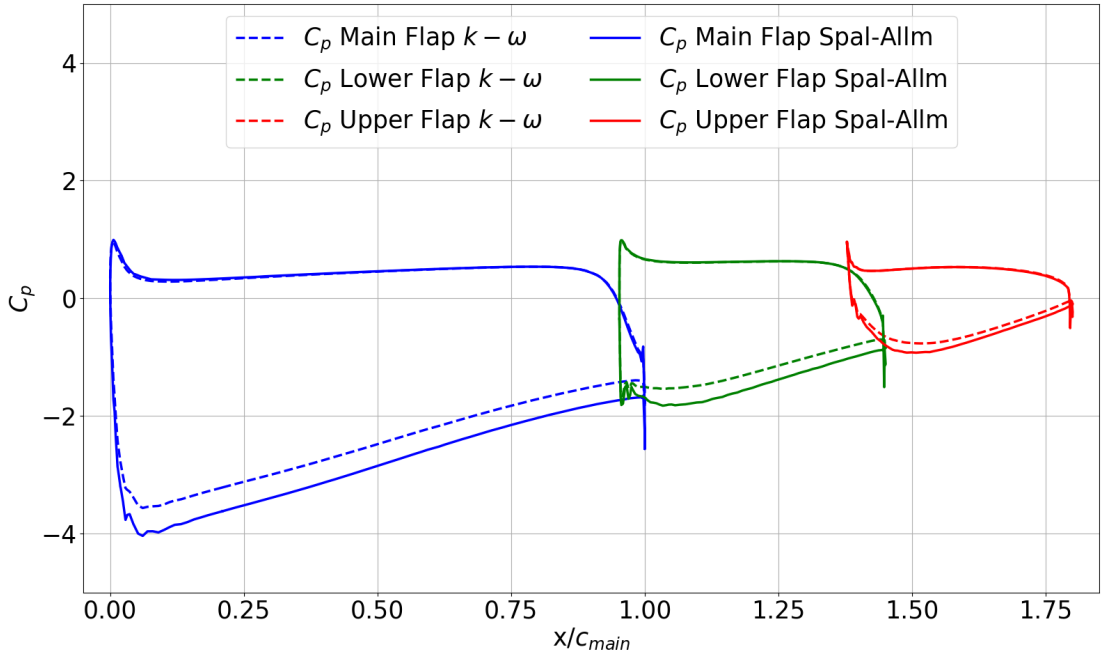


Figura 5.23: Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.63$

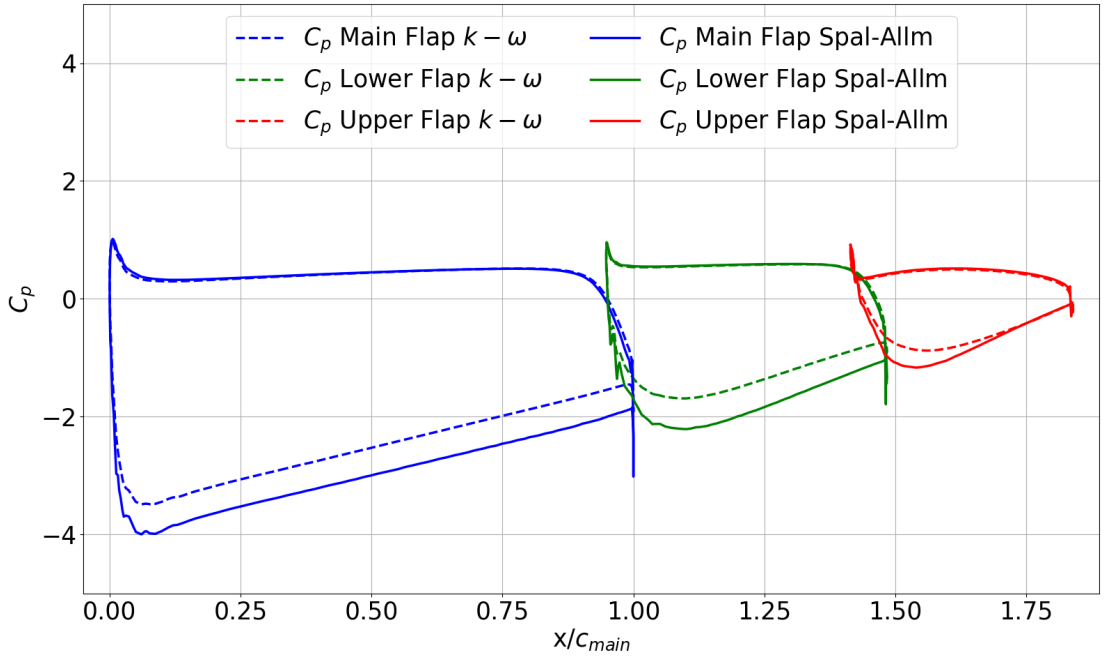


Figura 5.24: Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.74$

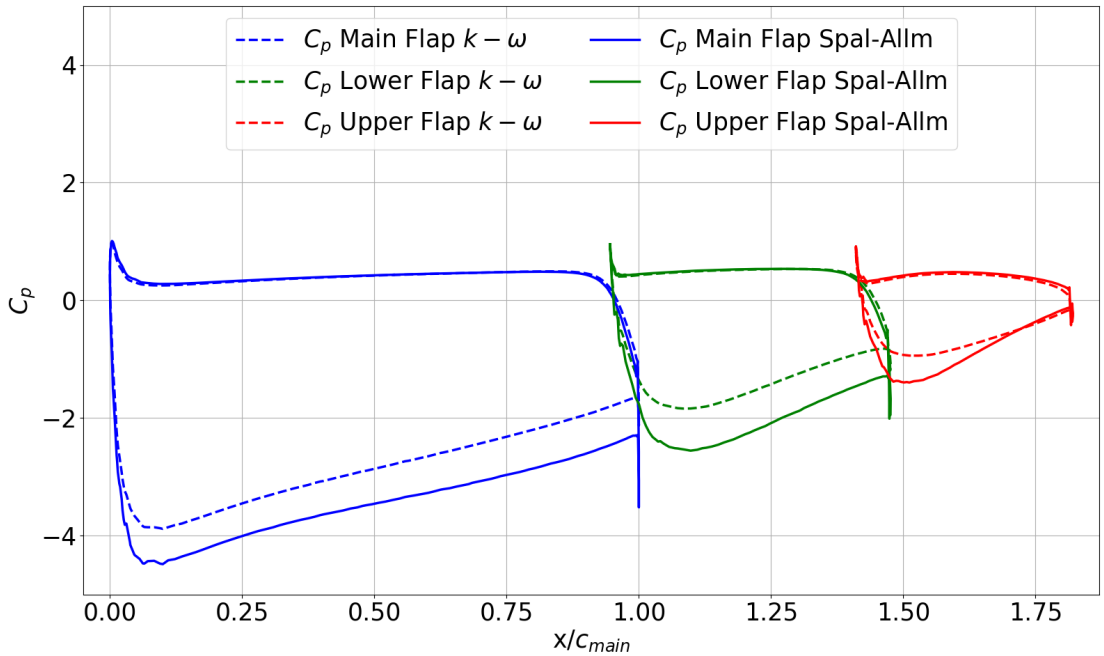


Figura 5.25: Confronto del C_p tra modello $k-\omega$ SST e Spalart-Allmaras - $y/s = 0.79$

Capitolo 6

Metodo dell'aggiunto

All'interno di questo capitolo verrà descritto il metodo dell'aggiunto [17] e come è stato applicato in questo lavoro di tesi con l'obiettivo di ricavare una mappa di sensitività dell'ala e successivamente ottimizzarne la forma rispetto ad una funzione obiettivo prestabilita. Verranno inoltre mostrate le modifiche da applicare al setup del calcolo diretto (visibile al Capitolo 4), per un corretto svolgimento del calcolo dell'aggiunto.

6.1 Analisi della sensitività e metodo dell'aggiunto

Tra i vari obiettivi di questo lavoro di tesi ci sono il calcolo e l'analisi della mappa di sensitività della geometria studiata e la successiva ottimizzazione di forma di questa.

Cercando di rimanere in tema con questo elaborato la domanda che ci possiamo porre è: una volta definita una quantità di interesse come ad esempio una forza esercitata sull'ala, questa, quanto è sensibile alla variazione della geometria dell'ala stessa?

La sensitività è quindi la variazione che si ottiene nell'output di un certo modello matematico misurata rispetto a delle variazioni negli input del modello stesso. Mettiamo caso che qui l'output sia "$J(u)$", l'input "u" e la variazione dell'input sopra citata "δu". Sarà possibile scrivere che:

$$\delta J(u) = \frac{\partial J(u)}{\partial u}\delta u \tag{6.1}$$

Dove "$\delta J(u)$" è la variazione dell'output del modello matematico definita come:

$$\delta J(u) = J(u + \delta u) - J(u) \tag{6.2}$$

In particolare se "u" è un vettore, mettiamo caso la posizione di un punto su una geometria, allora il termine "$\frac{\partial J(u)}{\partial u}$" è un gradiente ed è quindi possibile in un processo di ottimizzazione parlare di "ottimizzazione basata sul gradiente". Il gradiente fornisce informazioni sulla direzione di variazione dell'input, quindi su come modificare la geometria. Quindi la nuova variabile di input può essere ricavata attraverso la formulazione:

$$u_{i+1} = u_i - \rho\frac{\partial J(u)}{\partial u} \tag{6.3}$$

Immaginando sempre che il nostro input sia la geometria dell'ala otteniamo una nuova geometria (u_{i+1}) a partire dalla geometria iniziale (u) a cui viene sottratto il gradiente "$\frac{\partial J(u)}{\partial u}$" moltiplicato per un valore "ρ" detto passo.

L'obiettivo a questo punto è di calcolare questa quantità in un modo accurato ed efficiente per una serie di problemi sia teorici che pratici.

Ipotizziamo di stare lavorando con dei vettori, allora un metodo particolarmente efficiente nel momento in cui le dimensioni della variabile di input "u" sono notevolmente maggiori di quella di output "$J(u)$" è il metodo dell'aggiunto.
Consideriamo un problema di ottimizzazione d'esempio in cui "ϕ", "c" e "g" sono tre vettori di dimensione N, "c" è noto, "g" è la nostra variabile di input ed "A" è una matrice nota. Possiamo scrivere che:

$$J(\phi, g) = c^T \phi \tag{6.4}$$

Ricaviamo il valore di "ϕ" attraverso l'equazione:

$$A\phi = g \quad \Rightarrow \quad \phi = A^{-1} g \tag{6.5}$$

A questo punto è possibile ottenere "J" e, considerando un classico algoritmo di ottimizzazione che ricerca il minimo della funzione appena ricavate, è possibile scrivere che:

$$g_{i+1} = g_i - \rho(\frac{\partial J}{\partial g})_i \tag{6.6}$$

Attraverso questa equazione viene perturbato "g" e si risolve nuovamente la (6.4) e la (6.5) fino a che non si arriva al punto di minimo dove il gradiente della funzione obiettivo è pari a 0 e $g_{i+1} = g_i$.
Ci rimane a questo punto da capire come calcolare il gradiente della funzione "J". Un modo per farlo è utilizzare il metodo delle differenze finite, il gradiente di "J" rispetto a "g" viene quindi calcolato come:

$$\frac{\partial J}{\partial \vec{g}} \cdot \vec{e}_n = \frac{J(\boldsymbol{\phi}, \vec{g} + \epsilon \vec{e}_n) - J(\boldsymbol{\phi}, \vec{g})}{\epsilon} \tag{6.7}$$

Dove si ha $n = 1, ..., N$, $\epsilon \to 0^+$ e $\vec{e}_n$ è un vettore di modulo unitario. Questo procedimento avrà un costo computazionale pari a $N + 1$, questo significa che otteniamo una perturbazione per ogni indice "n" e dovremo risolvere l'equazione (6.5) $N + 1$ volte.

Introduciamo invece ora nuovo sistema lineare (la nostra equazione aggiunta) tale per cui:

$$A^T a = c \tag{6.8}$$

Dove "a" ha le stesse dimensioni di "ϕ". Se andiamo a sostituire "c" nella (6.4) si ottiene quindi:

$$J = c^T \phi = (A^T a)^T \phi = a^T A \phi = a^T g \tag{6.9}$$

A questo punto "J" dipende direttamente dal vettore "g" e possiamo scrivere che:

$$\frac{\partial J}{\partial g} = a \tag{6.10}$$

Il costo computazionale quindi in questo caso è pari a 1 e soprattutto è indipendente da "N", quindi la soluzione non dipende dalle dimensioni del vettore di input "g".

6.1.1 L'operatore aggiunto

L'operatore aggiunto viene introdotto da Lagrange nel XVIII secolo quando non esistevano ancora i computer e tutto veniva risolto analiticamente. A partire da un'equazione differenziale lineare del tipo $Au = f$ ha introdotto l'equazione aggiunta $A^*u^* = g$, dove "A^*" viene definito come operatore aggiunto e "u^*" variabile aggiunta, da cui ha ottenuto l'uguaglianza (6.11) definita come "Uguaglianza di Lagrange":

$$\int u^* Au \, dx = \int uA^*u^* \, dx \tag{6.11}$$

valida nel caso in cui $(u, u^*) \to 0$ in corrispondenza degli estremi di integrazione, siccome però questa condizione non è sempre necessariamente vera è possibile scrivere la (6.11) utilizzando dei prodotti scalari, quindi nella forma:

$$\langle u^*, Au \rangle = \langle u, A^*u^* \rangle + B.T. \tag{6.12}$$

Dove $\langle ., . \rangle$ è un prodotto scalare. Il termine di destra dell'equazione viene ottenuto mediante un integrale per parti e "$B.T.$" è legato ai termini al contorno che potrebbero essere diversi da 0.

Bisogna sottolineare che in questo caso "A" è un operatore lineare ma, il problema originale potrebbe non esserlo. Nel caso in cui dovessimo definire l'operatore aggiunto di un operatore non lineare è necessario prima linearizzare l'operatore e successivamente definire il suo operatore aggiunto.

In termini di matrici la matrice aggiunta di una matrice "A" di dimensioni $m \times n$ e contenente elementi complessi è la matrice "A^*", ottenuta facendo la trasposta della matrice "A", quindi di dimensioni $n \times m$, e trasformando gli elementi complessi nei loro rispettivi complessi coniugati, si elimina quindi la parte immaginaria e si conserva quella reale. Si ha quindi che:

$$A^* = \overline{A^T} \tag{6.13}$$

Se si considera un sistema lineare discreto come quello visto precedentemente della forma $Ax = b$ dove "A" è una matrice di dimensioni $n \times n$ composta di valori reali e "x" e "b" sono due vettori di dimensione n e introduciamo una variabile aggiunta "y" è possibile definire l'aggiunto, rispetto a quanto visto nella (6.12) come:

$$\langle y, Ax \rangle = \langle A^*y, x \rangle \tag{6.14}$$

Assumiamo ora che l'aggiunto A^*y sia rappresentato da un vettore "c^n" e, ricordando che il prodotto scalare tra due vettori d'esempio "a" e "b" di uguale dimensione può essere scritto come $\langle a, b\rangle = a^T b$, sostituiamo nella (6.14) i valori "b" e "c" e otteniamo:

$$\langle y, b\rangle = y^T b = c^T x = \langle c, x\rangle \tag{6.15}$$

dove "y" può essere interpretata come la sensitività rispetto a "b". Tuttavia, non è stato specificato cosa sia o a cosa sia legato il vettore "c". L'aggiunto in se, quindi, non ha un vero e proprio significato fisico ma è utile in quanto strumento matematico per risolvere dei problemi specifici che lo hanno. Se leghiamo "c" con "J", come visto nel primo esempio, attraverso una formulazione del tipo $J = c^T x$, allora è possibile fornire una interpretazione fisica del risultato.

Aggiunto continuo e discreto

Le equazioni aggiunte possono essere derivate utilizzando due approcci differenti tra loro, da qui deriva la distinzione tra aggiunto continuo, quello implementato all'interno di OpenFOAM v2412, e aggiunto discreto, quello implementato in DAFoam, introdotto al Capitolo 2. Entrambi questi sistemi hanno vantaggi e svantaggi ovviamente. Nell'approccio continuo le equazioni aggiunte vengono derivate utilizzando le equazioni di Navier-Stokes prima che queste vengano discretizzate per essere risolte. Nell'approccio discreto invece le equazioni aggiunte vengono ricavate quando le equazioni dirette sono già state discretizzate.

Nel primo caso i vantaggi sono legati al fatto che la derivazione delle equazioni aggiunte è diretta ed è possibile costruire il solutore dell'aggiunto utilizzando parti di codice già esistenti, utilizzate per il solutore diretto. Gli svantaggi derivano dal fatto che le equazioni aggiunte una volta derivate da quelle continue vanno discretizzate e la precisione del risultato ottenuto dipende dallo schema di discretizzazione utilizzato. Tanto più viene aumentata la risoluzione spaziale e temporale secondo cui vengono discretizzate le equazioni aggiunte tanto più saranno precisi i risultati. Si hanno inoltre dei problemi legati alla complessità di implementazione delle condizioni al contorno.

Nel secondo caso invece le equazioni aggiunte vengono derivate a partire dalle equazioni dirette già discretizzate. Il vantaggio principale in questo approccio è che le equazioni aggiunte sono direttamente correlate al problema diretto e forniranno quindi risultati molto più precisi. Il livello di precisione raggiunto è infatti quello della macchina che si sta utilizzando per risolvere il problema. Inoltre, essendo direttamente legate le equazioni aggiunte alla forma discreta delle equazioni dirette, il loro livello di precisione non dipende dalla risoluzione spaziale e temporale della discretizzazione. Lo svantaggio in questo caso sta nel fatto che la derivazione delle equazioni aggiunte è particolarmente complicata e anche l'implementazione a livello di codice risulta più difficile, se confrontata con l'approccio continuo, in quanto non si può utilizzare parte del codice già esistente.

6.2 Equazioni aggiunte applicate al caso di Navier-Stokes

Per spiegare il modello matematico si parte dalle equazioni di Navier-Stokes mediate (RANS):

$$\begin{cases} \nabla \cdot U = 0 \\ \rho \dfrac{DU}{Dt} = -\nabla(P + \dfrac{2}{3}\rho k) + \nabla \mu_{eff}(\nabla U + \nabla U^T) \end{cases} \tag{6.16}$$

Ipotizzando di trovarsi in un flusso incomprimibile e stazionario ($U = \vec{u}$ e $P = p$) e ricordando che $(\nabla U + \nabla U^T) = 2S_R$ e $\nu = \frac{\mu}{\rho}$ possono essere riscritte come:

$$\begin{cases} \nabla \cdot \vec{u} = 0 \\ (\vec{u} \cdot \nabla)\vec{u} = -\nabla p + \nabla \cdot 2\nu_{eff} S_R \end{cases} \tag{6.17}$$

Con $\nu_{eff} = \nu + \nu_t$. Il problema di ottimizzazione può essere visto come la volontà di minimizzare una funzione obiettivo $J = J(\alpha, \vec{u}, p)$ con $\vec{u}$ e p velocità e pressione, cioè le variabili dirette, e α la variabile di design. Il vincolo alla minimizzazione di J è quindi dato dalle equazioni RANS che devono essere ovviamente rispettate. Per legare tra loro funzione obiettivo e vincolo dato dalle equazioni di Navier-Stokes si definisce la Lagrangiana tale per cui:

$$L = J + \int_\Omega \langle (\vec{v}, q), R \rangle \, d\Omega \tag{6.18}$$

Dove "$\vec{v}$" e "q" sono rispettivamente campo di velocità e di pressione dell'aggiunto introdotti come moltiplicatori di Lagrange, con "R" vengono indicate le equazioni RANS e il tutto viene integrato sul dominio di calcolo "Ω". Questa notazione indica che ogni vincolo viene moltiplicato per i moltiplicatori di Lagrange e i vari contributi vengono sommati tra loro su tutto il dominio. Viene studiato il caso in cui la variazione totale di "L", rispetto alle variabili dirette, aggiunte e alla variabile di design, è uguale a zero trattando le variabili come indipendenti:

$$\delta L = \delta_{\vec{v}} L + \delta_q L + \delta_\alpha L + \delta_{\vec{u}} L + \delta_p L = 0 \tag{6.19}$$

Nell'equazione (6.19) ogni termine "$\delta_x L$" è pari a "$\frac{\partial L}{\partial x}\delta x$". La derivazione dell'equazione aggiunta consiste nel imporre che ogni gradiente "$\frac{\partial L}{\partial x}$" sia pari a 0, indipendentemente, data una variazione arbitraria "δx". Dato che il vettore "R" è composto da equazioni differenziali, la derivazione viene fatta attraverso una integrazione per parti. I gradienti $\frac{\partial L}{\partial q} = 0$ e $\frac{\partial L}{\partial \vec{v}} = 0$ sostituiscono le equazioni RANS. Da $\frac{\partial L}{\partial p} = 0$ e $\frac{\partial L}{\partial \vec{u}} = 0$ si ricavano le equazioni aggiunte e da $\frac{\partial L}{\partial \alpha} = 0$ si ricava l'espressione della sensitività.

Dopo ulteriori derivazioni si arriva alla forma finale delle equazioni di Navier-Stokes aggiunte:

$$\begin{cases} -2S_R(\vec{v})\vec{u} = -\nabla q + \nabla \cdot (2\nu_{eff} S_R(\vec{v})) - \alpha \vec{v} - \dfrac{\partial J_\Omega}{\partial \vec{u}} \\ \nabla \cdot \vec{v} = \dfrac{\partial J_\Omega}{\partial p} \end{cases} \tag{6.20}$$

Troviamo il termine "J_Ω" in quanto è possibile suddividere la funzione obiettivo "J" in due parti tali per cui:

$$J = \int_\Omega J_\Omega \, d\Omega + \int_\Gamma J_\Gamma \, d\Gamma \tag{6.21}$$

il termine "J_Ω" va a 0 se nella funzione obiettivo non vi è nessun contributo da parte del flusso presente nel dominio. Le variazioni di quantità di moto e pressione aggiunte entrano a far parte del solutore solo attraverso le condizioni al contorno che possono essere ricavate a partire dalle (6.21) e scritte come:

$$\begin{cases} \displaystyle\int_\Gamma \left(\vec{n}(\vec{v}\cdot\vec{u}) + \vec{v}(\vec{u}\cdot\vec{n}) + 2\nu_{eff}\vec{n}\cdot D(\vec{v}) - q\vec{n} + \frac{\partial J_\Gamma}{\partial \vec{u}} \right) \delta\vec{u} \, d\Gamma - \int_\Gamma 2\nu_{eff}\vec{n}\cdot D(\delta\vec{u})\cdot\vec{v} \, d\Gamma = 0 \\ \displaystyle\int_\Gamma \left(\vec{v}\cdot\vec{n} + \frac{\partial J_\Gamma}{\partial p} \right) \delta p \, d\Gamma = 0 \end{cases} \tag{6.22}$$

A questo punto è possibile inserire una ulteriore semplificazione legata al valore di "ν_t" che può essere posto uguale a 0, questo fa si che la variazione di viscosità effettiva "$\delta\nu_{eff}$" venga trascurata nel momento in cui si vanno a ricavare le equazioni aggiunte. Questa situazione è sicuramente vera per un caso in cui si studia un flusso laminare, per i casi caratterizzati da un flusso turbolento invece prende il nome di "frozen turbulence". Questa semplificazione permette di ridurre il costo computazionale dell'equazione aggiunta a scapito ovviamente della precisione dei risultati ottenuti. Inoltre, è importante sottolineare che velocità e pressione aggiunta vengono così definite e denominate per semplificare la comprensione di quanto detto fin'ora associandole a velocità e pressione del calcolo diretto ma, come detto precedentemente, non hanno alcun significato fisico, quindi non vanno interpretate come si interpretano solitamente velocità e pressione [17] [18] [19].

6.3 Metodo dell'aggiunto: setup di OpenFOAM

Per risolvere le equazioni aggiunte all'interno di OpenFOAM è necessario utilizzare un setup specifico. Per impostare correttamente il calcolo è stato seguito il manuale per l'utilizzo di "adjointOptimisationFoam", l'applicazione utilizzata per la risoluzione dell'aggiunto, scritto dal gruppo di "Parallel CFD & Optimisation Unit" del Politecnico di Atene [20]. Anche in questo caso come per il calcolo diretto vengono utilizzate tre directory principali:

- Directory "0"
- Directory "constant"
- Directory "system"

all'interno delle quali verranno fatte delle aggiunte e delle modifiche rispetto a quanto visto nel Capitolo 4.

6.3.1 Directory "0"

All'interno della directory "0" devono essere inseriti tutti i file relativi alle condizioni al contorno o condizioni iniziali dell'aggiunto, oltre a quelle per il calcolo diretto descritte nel Capitolo 4. Per far si che questo converga è estremamente importante che alle condizioni al contorno delle grandezze dirette vengano associate le condizioni al contorno delle grandezze aggiunte corrette, se questo non viene fatto ci sono infatti forti probabilità che il calcolo dell'aggiunto non vada a convergenza. Dovremo quindi aggiungere i seguenti file:

- Velocità aggiunta "Ua"
- Pressione aggiunta "pa"
- Energia cinetica turbolenta aggiunta "ka"
- Tasso di dissipazione specifico aggiunto "wa"
- Viscosità turbolenta modificata aggiunta "nuaTilda" (solo per Spalart-Allmaras)

che verranno ora visualizzati nel dettaglio. La struttura dei file è la medesima utilizzata per il calcolo diretto visibile nel Capitolo 4, verranno solo riportate all'interno di alcune tabelle le associazioni tra condizione al contorno del diretto e condizione al contorno dell'aggiunto.

Velocità aggiunta "Ua"

Nome patch	Condizione al contorno - diretto	Condizione al contorno - aggiunto
p-outlet	*inletOutlet*	*adjointFarFieldVelocity*
s-domain-limit-lat	*slip*	*slip*
s-domain-limit-sup	*slip*	*slip*
v-inlet	*fixedValue*	*adjointInletVelocity*
w00-ground	*fixedValue*	*adjointWallVelocity*
w00np-fwl	*slip*	*slip*
"wing-patches"	*fixedValue*	*adjointWallVelocity*
"wheel-patches"	*rotatingWallVelocity*	*adjointRotatingWallVelocity*

Tabella 6.1: Condizioni al contorno della velocità di diretto e aggiunto

Le uniche due condizioni al contorno su cui è necessario fare commenti sono *adjointWallVelocity* e *adjointFarFieldVelocity*. La prima è la condizione specifica da associare alla velocità aggiunta su una specifica porzione del dominio nel momento in cui sulla stessa patch nelle condizioni al contorno di "U" è presente "fixedValue" e nelle condizioni al contorno di "p" si trova la tipologia "zeroGradient". La condizione al contorno *adjointFarFieldVelocity*, invece, è specifica per i casi in cui su una patch del dominio viene utilizzata su "U" la condizione al contorno "inletOutlet".

Pressione aggiunta "pa"

Nome patch	Condizione al contorno - diretto	Condizione al contorno - aggiunto
p-outlet	*fixedValue*	*adjointOutletPressure*
s-domain-limit-lat	*slip*	*slip*
s-domain-limit-sup	*slip*	*slip*
v-inlet	*zeroGradient*	*zeroGradient*
w00-ground	*zeroGradient*	*zeroGradient*
w00np-fwl	*slip*	*slip*
"wing-patches"	*zeroGradient*	*zeroGradient*
"wheel-patches"	*zeroGradient*	*zeroGradient*

Tabella 6.2: Condizioni al contorno della pressione di diretto e aggiunto

Si osserva in questo caso come le condizioni di diretto e aggiunto siano le stesse eccetto per la patch relativa all'outlet, dove viene utilizzata la condizione al contorno specifica per la pressione aggiunta *adjointOutletPressure*.

Energia cinetica turbolenta aggiunta "ka"

Nome patch	Condizione al contorno - diretto	Condizione al contorno - aggiunto
p-outlet	*zeroGradient*	*adjointOutletKa*
s-domain-limit-lat	*slip*	*slip*
s-domain-limit-sup	*slip*	*slip*
v-inlet	*fixedValue*	*adjointZeroInlet*
w00-ground	*kqRWallFunction*	*kaqRWallFunction*
w00np-fwl	*slip*	*slip*
"wing-patches"	*kqRWallFunction*	*kaqRWallFunction*
"wheel-patches"	*kqRWallFunction*	*kaqRWallFunction*

Tabella 6.3: Condizioni al contorno dell'energia cinetica turbolenta di diretto e aggiunto

Tasso di dissipazione specifico aggiunto "wa"

Nome patch	Condizione al contorno - diretto	Condizione al contorno - aggiunto
p-outlet	*zeroGradient*	*adjointOutletWa*
s-domain-limit-lat	*slip*	*slip*
s-domain-limit-sup	*slip*	*slip*
v-inlet	*fixedValue*	*adjointZeroInlet*
w00-ground	*omegaWallFunction*	*waWallFunction*
w00np-fwl	*slip*	*slip*
"wing-patches"	*omegaWallFunction*	*waWallFunction*
"wheel-patches"	*omegaWallFunction*	*waWallFunction*

Tabella 6.4: Condizioni al contorno del tasso di dissipazione specifico di diretto e aggiunto

Viscosità turbolenta modificata aggiunta "nuaTilda"

Questa viene utilizzata al posto di "ka" e "wa" nel caso in cui venisse utilizzato il modello di turbolenza Spalart-Allmaras.

Nome patch	Condizione al contorno - diretto	Condizione al contorno - aggiunto
p-outlet	*zeroGradient*	*adjointOutletNuaTilda*
s-domain-limit-lat	*slip*	*slip*
s-domain-limit-sup	*slip*	*slip*
v-inlet	*fixedValue*	*adjointInletNuaTilda*
w00-ground	*fixedValue*	*fixedValue*
w00np-fwl	*slip*	*slip*
"wing-patches"	*fixedValue*	*fixedValue*
"wheel-patches"	*fixedValue*	*fixedValue*

Tabella 6.5: Condizioni al contorno della viscosità turbolenta modificata di diretto e aggiunto

6.3.2 Directory "constant"

All'interno della directory "constant" vengono inseriti due file supplementari: "adjointRASProperties" e "dynamicMeshDict". Il primo file contiene le informazioni a proposito del modello di turbolenza utilizzato per risolvere l'aggiunto. Al suo interno sono presenti due dictionary, il primo, "adjointRASModel", ci indica il modello di turbolenza che vogliamo utilizzare per risolvere l'aggiunto. Il secondo, "adjointTurbulence", permette di scegliere se comprendere o meno all'interno della risoluzione dell'aggiunto le equazioni relative alle grandezze caratteristiche della turbolenza per il modello che stiamo utilizzando. Di seguito è possibile osservare un esempio della struttura di questo file per il modello di turbolenza $k-\omega$ SST.

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      adjointTurbulenceProperties;
}
adjointRASModel adjointkOmegaSST;
adjointTurbulence on;
```

A seconda del modello di turbolenza che si vuole utilizzare è quindi necessario cambiare "adjointRASModel", per il lavoro di tesi esposto nel seguente elaborato sono stati utilizzati due modelli di turbolenza dell'aggiunto: "adjointkOmegaSST" e "adjointSpalartAllmaras". Inoltre, a causa di problemi legati al non raggiungimento della convergenza durante il calcolo dell'aggiunto utilizzando "adjointkOmegaSST" e per verificare che tutte le condizioni al contorno fossero corrette, è stato utilizzato anche "adjointLaminar" che risolve l'aggiunto come se si trattasse di un caso laminare. Nel Capitolo 7 verranno esposte le differenze tra i vari risultati ottenuti utilizzando le diverse impostazioni.

Il file "dynamicMeshDict" viene invece utilizzato nel momento in cui si effettuano i cicli di ottimizzazione andando a variare la forma della geometria in esame, verrà quindi spiegato più nel dettaglio nel capitolo legato al processo di ottimizzazione (Capitolo 7).

6.3.3 Directory "system"

Per effettuare correttamente il calcolo dell'aggiunto la prima cosa da fare è, nel file "controlDict", sostituire l'applicazione utilizzata passando da "simpleFoam" ad "adjointOptimisationFoam", successivamente è necessario, sempre della directory "system", aggiungere il file "optimisationDict" che gestisce il processo di calcolo diretto e aggiunto e, a seconda delle impostazioni utilizzate, il calcolo della mappa di sensitività o la deformazione della geometria analizzata.

File "optimisationDict"

All'interno del file "optimisationDict" nella directory "system" sono contenute tutte le informazioni relative a: tipo di processo di ottimizzazione da eseguire, risoluzione del calcolo diretto, risoluzione dell'aggiunto, funzione obiettivo utilizzata e calcolo delle mappe di sensitività.

Il primo sottogruppo che troviamo all'interno di questo file è "optimisationManager", questo determina il tipo di processo che vogliamo eseguire: nel caso in cui si utilizzi la funzione "singleRun" calcolo diretto e aggiunto vengono risolti una sola volta e viene calcolata la mappa di sensitività rispetto alla funzione obiettivo imposta. Nel caso in cui, invece, venga utilizzata la funzione "steadyOptimisation" è necessario inserire delle ulteriori impostazioni specifiche all'interno del file. Dopo aver risolto calcolo diretto e aggiunto, infatti, viene modificata la forma della geometria, viene poi effettuato un check della mesh, secondo le istruzioni riportate nel file "checkMeshDict", e successivamente ricalcolata la soluzione del diretto e dell'aggiunto e così via per tanti quanti sono gli step di ottimizzazione desiderati o fino a quando non si raggiungono le condizioni di convergenza imposte sull'andamento della funzione obiettivo che si sta osservando.

Considerando un caso in cui viene utilizzata la funzione "singleRun" la parte del file "optimisationDict" relativa alla risoluzione del calcolo diretto ha una forma del tipo:

```
optimisationManager singleRun;
primalSolvers
{
    op1
    {
        active                  true;
        type                    incompressible;
        solver                  simple;
        solutionControls
        {
            consistent yes;
            nIters 6000;
            residualControl
            {
                "p.*"       1.e-5;
                "U.*"       1.e-5;
            }
            averaging
            {
                average     true;
                startIter   5000;
            }
        }
    }
}
```

Si osservano una serie di impostazioni quali: "active true;" che sta a significare che nel momento in cui viene lanciato il calcolo viene effettuato il calcolo diretto, il flusso di tipo incomprimibile e il solver settato su simple, quindi il calcolo viene effettuato secondo le impostazioni proprie di "simpleFoam", di cui si è parlato nel Capitolo 4. Successivamente troviamo una parte di controllo in cui viene inserito il numero di iterazioni per le quali si vuole che il calcolo venga eseguito, "nIters". Questo, a meno di non raggiungere prima le condizioni di convergenza imposte sui residui di pressione e velocità in "residualControl".

Nel dictionary "averaging" sono contenute le impostazioni per mediare i risultati del calcolo diretto su un numero di iterazioni definito dall'utente, questo è molto importante quando si lavora con l'aggiunto. Per sperare di ottenere una buona convergenza dell'aggiunto, infatti, è necessario avere un risultato del calcolo diretto molto stabile. In una situazione nella quale questo non avviene, come nel caso qui esaminato, è necessario quindi mediare la soluzione del diretto su molte iterazioni per ottenere un risultato stabile che sia una base adatta per la successiva risoluzione dell'aggiunto.

Il dictionary "adjointManagers" contiene invece le informazioni relative al calcolo dell'aggiunto, la struttura è molto simile a "primalSolvers" appena analizzato ma si differenzia per alcuni punti. Innanzitutto, basandosi il calcolo dell'aggiunto sui risultati del calcolo diretto è necessario indicare attraverso "primalSolvers op1;" il calcolo diretto a cui bisogna fare riferimento. Il solutore dell'aggiunto in questo caso è "adjointSimple", l'unico implementato e disponibile in OpenFOAM v2412, successivamente troviamo il dictionary "objectives", probabilmente uno dei più importanti fino ad ora visto lo scopo del lavoro di tesi. All'interno di "objectives", infatti, sono contenute le informazioni riguardo la funzione obiettivo rispetto alla quale si vuole calcolare la mappa di sensitività e successivamente eseguire l'ottimizzazione di forma. Nel nostro caso sono state utilizzate come funzioni obiettivo unicamente forze, ma all'interno di OpenFOAM sono implementate diverse funzioni obiettivo a seconda dello scopo che si vuole perseguire. La forma della funzione obiettivo relativa alle forze implementata in OpenFOAM è:

$$J = \frac{\int_{Sw} \rho(-\tau_{ij} n_j + p n_i) r_i dS}{\frac{1}{2}\rho A U^2} \tag{6.23}$$

ed è inoltre possibile fare una combinazione di più funzioni obiettivo elencandole una dopo l'altra nel dictionary "objectiveNames", la funzione obiettivo finale avrà una forma del tipo:

$$J = \sum_i w_i J_i \tag{6.24}$$

Dove "J_i" è una singola funzione obiettivo e "w_i" il peso ("weight" al'interno dei vari dictionary) associato a questa. OpenFOAM di default quando si esegue un processo di ottimizzazione lavora sulla minimizzazione della funzione obiettivo, se si volesse invece massimizzarla bisogna porre il peso di questa pari a -1 oppure, nel caso delle forze, invertire il vettore lungo la quale viene calcolata la forza.

Inoltre, è possibile inserire all'interno delle funzioni obiettivo dei valori target di queste. Per farlo è necessario inserire nel dictionary "adjS1" una riga con "isConstraint true;" ed inserire il valore target nel sottogruppo della funzione obiettivo. Il problema in questo caso è che se si volesse fare una funzione obiettivo composta da una funzione obiettivo con valore target ed una no, servirebbe calcolare l'aggiunto due volte. Questo, comporta un costo sotto il punto di vista computazionale doppio.

Nella pagina successiva è possibile osservare la parte di script relativa a quanto appena detto per una più facile comprensione.

```
adjointManagers
{
    adjManager1
    {
        primalSolver            op1;
        adjointSolvers
        {
            adjS1
            {
                active                  true;
                type                    incompressible;
                solver                  adjointSimple;
                computeSensitivity      true;
                objectives
                {
                    type incompressible;
                    objectiveNames
                    {
                        lift
                        {
                            weight      1;
                            type        force;
                            patches     ("wing-patches");
                            direction   (0 0 1);
                            Aref        1;
                            rhoInf      1.225;
                            UInf        50;
                        }
                    }
                }
                ATCModel
                {
                    ATCModel          cancel;
                }
                solutionControls
                {
                    ...
                }
            }
        }
    }
}
```

Rimane infine la parte relativa al calcolo della mappa di sensitività, nel dictionary "optimisation" sono contenute tutte le informazioni necessarie. Rispetto ad una singola soluzione dell'aggiunto possono essere calcolate diverse tipologie di sensitività attraverso "sensitivityType multiple;", in questo caso ne viene riportata solo una per semplicità, quella poi utilizzata per produrre le rappresentazioni. Importante in questo caso notare il comando "smoothSensitivities true;", infatti nei casi industriali come quello in esame è molto probabile ottenere mappe di sensitività poco interpretabili a livello visivo a causa di forti disuniformità nella mappa stessa. Questa problematica può essere attenuata in due modi, il primo è quello di inserire il valore denominato "meanRadiusMultiplier", la sensitività in un punto sarà quindi data dalla media delle sensitività dei punti contenuti in una sfera con raggio definito dall'utente, nel nostro caso 10. Il secondo metodo consiste nel settare "smoothSensitivities true;", questo richiede di aggiungere due ulteriori file nella directory "system": "faSchemes" e "faSolution", con l'obiettivo di uniformare la mappa di sensitività e fornire innanzitutto una visualizzazione meglio interpretabile e in secondo luogo ottenere una deformazione uniforme e più continua possibile della geometria, soprattutto in casi in cui dovessero essere utilizzati molti punti di controllo.

```
optimisation
{
    designVariables
    {
        type shape;
        shapeType volumetricBSplines;
        sensitivityType multiple;
        sensitivityTypes
        (
            faceBased-RMult_10
        );
        patches ("wing-patches");
        faceBased-RMult_10
        {
            sensitivityType             surface;
            patches                     ("wing-patches");
            includeSurfaceArea          true;
            smoothSensitivities         true;
            meanRadiusMultiplier        10;
            suffix                      Rmult10;
            iters                       5000;
        }
    }
}
```

File "fvSchemes" e "fvSolution"

I file "fvSchemes" e "fvSolution" sono strutturati per il calcolo dell'aggiunto esattamente come quelli del calcolo diretto. Bisognerà quindi definire gli schemi di discretizzazione per le equazioni aggiunte in "fvSchemes". In "fvSolution" sono poi contenute le informazioni relative ai metodi di risoluzione delle equazioni aggiunte discretizzate e gli URF, che devono essere ovviamente definiti anche per i campi dell'aggiunto. All'interno del Capitolo 7 sono contenute maggiori informazioni a proposito dei setup di calcolo dell'aggiunto.

Capitolo 7

Ottimizzazione

All'interno di questo capitolo verranno mostrati i risultati ricavati dal calcolo dell'aggiunto a partire dai risultati dei calcoli diretti ottenuti con i due modelli di turbolenza $k-\omega$ SST e Spalart-Allmaras. Verranno mostrate e confrontate tra loro le mappe di sensitività e infine verrà analizzato il processo di ottimizzazione finale, l'effettiva deformazione della geometria ottenuta e un confronto tra i risultati ottenuti dal calcolo diretto prima e dopo la deformazione.

7.1 Primi risultati

L'idea iniziale del progetto era utilizzare DAFoam per calcolare la sensitività e procedere poi con l'ottimizzazione di forma, tuttavia, essendo l'aggiunto implementato in DAFoam di tipo discreto necessita, per griglie di calcolo con un grande numero di elementi, di una quantità di RAM che in questo lavoro di tesi non era accessibile, ne con i mezzi presenti in università e nemmeno con quelli presso Dallara. Per questo motivo anziché DAFoam si è deciso di utilizzare OpenFOAM in versione ESI, in particolare l'ultimo aggiornamento disponibile durante lo svolgimento dei calcoli, vale a dire OpenFOAM v2412. Su questa versione come sulle precedenti è implementato un aggiunto di tipo continuo, questo è ovviamente meno accurato dell'aggiunto discreto ma permette di utilizzare un quantitativo di RAM simile a quello utilizzato per i calcoli stazionari "classici" come descritto nel Capitolo 6 relativo all'aggiunto.

Sono stati riscontrati diversi problemi per quanto riguarda la convergenza del calcolo dell'aggiunto, probabilmente legati all'instazionarietà della soluzione iniziale. Per cercare di attenuare questa problematica la soluzione primaria è stata mediata sulle ultime 1000 iterazioni, in modo da "fornire" all'aggiunto delle condizioni di flusso più stabili possibili. Nella pagina successiva in figura 7.1 e figura 7.2 è possibile osservare rispettivamente residui e continuità del calcolo diretto effettuato utilizzando $k-\omega$ SST, è possibile osservare come i residui siano caratterizzati da valori elevati e da un comportamento molto oscillatorio.

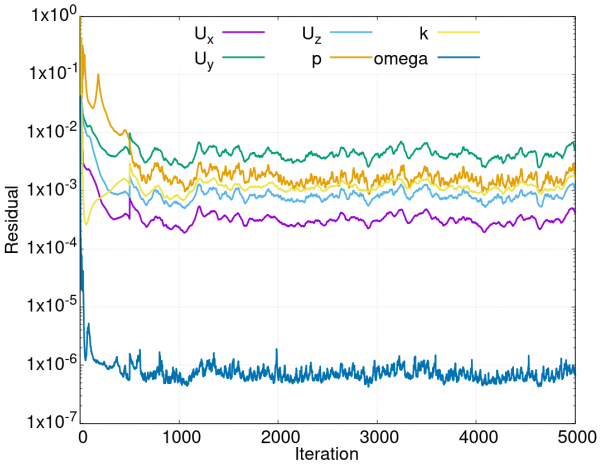


Figura 7.1: Residui calcolo diretto $k-\omega$

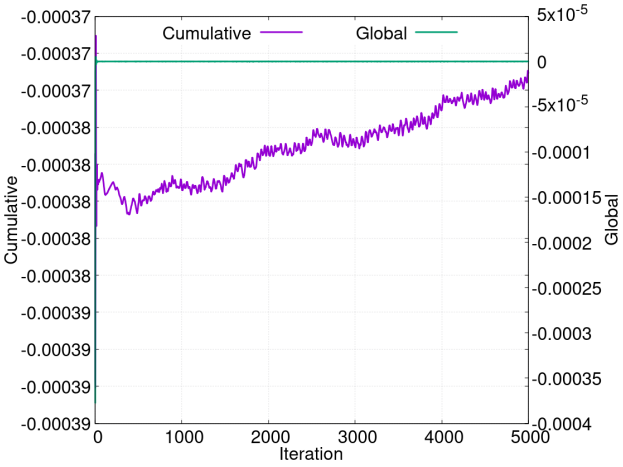


Figura 7.2: Continuità calcolo diretto $k-\omega$

Siccome l'aggiunto è vincolato alla soluzione diretta del calcolo è necessario che i risultati siano più stabili possibili. Nel nostro caso, a causa della forte instazionarietà del calcolo diretto, non è stato possibile ottenere una convergenza dell'aggiunto utilizzando come modello di turbolenza dello stesso "adjointkOmegaSST". Sono stati effettuati molti tentativi per risolvere questa situazione, come variare condizioni al contorno dell'aggiunto, schemi di discretizzazione, solver e fattori di rilassamento. Tuttavia, come detto, nessuno ha portato a dei risultati soddisfacenti. In figura 7.3 e 7.4 è possibile osservare il tipico andamento di residui e continuità dell'aggiunto, ottenuti utilizzando "adjointkOmegaSST".

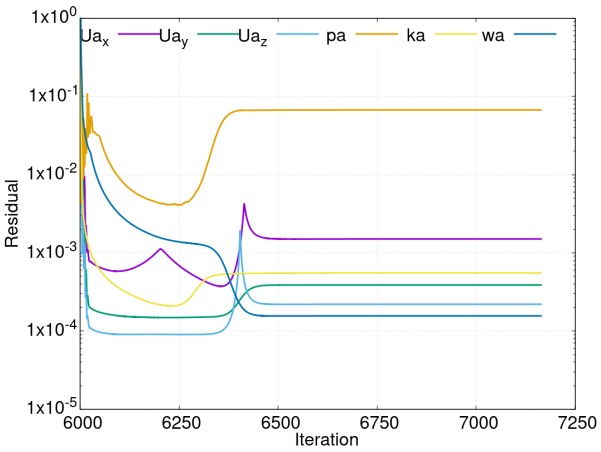


Figura 7.3: Residui calcolo aggiunto $k-\omega$

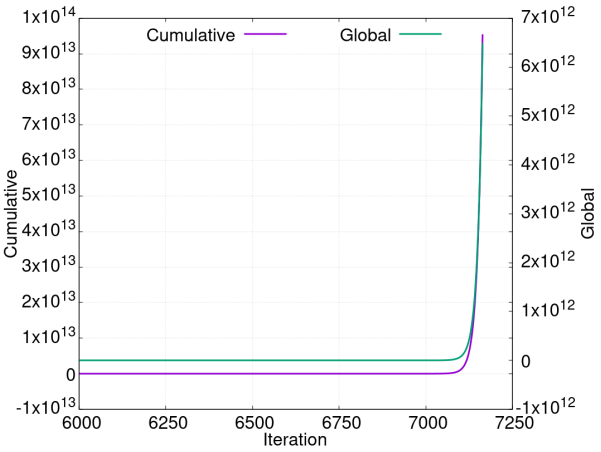


Figura 7.4: Continuità calcolo aggiunto $k-\omega$

7.1.1 Risoluzione dei problemi

Per risolvere i problemi incontrati nel cercare di ottenere una buona convergenza è stato chiesto un parere al team del Politecnico di Atene responsabile dello sviluppo dell'aggiunto continuo implementato in OpenFOAM v2412. I consigli per tentare di ottenere una buona convergenza sono stati i seguenti:

- Cercare una stabilizzazione del calcolo diretto, per esempio utilizzando un modello di turbolenza differente come Spalart-Allmaras. Questo, fornisce tendenzialmente risultati più stabili del calcolo diretto, quindi, almeno in linea di principio, dovrebbe portare ad una migliore convergenza dell'aggiunto.
- Controllare le condizioni al contorno dell'aggiunto perché, come si evince da quanto detto al Capitolo 6, è di fondamentale importanza che siano associate nel modo corretto alle condizioni al contorno del calcolo diretto per ottenere una buona convergenza.
- Utilizzare dei fattori di rilassamento (URF) per l'aggiunto specifici, in particolare:
  - $Ua = 0,7$
  - $pa = 0.1$
  - $ka = 0.2$
  - $wa = 0.2$
  - $nuaTilda = 0.2$ (Se si utilizza Spalart-Allmaras)
- Provare a ricavare una soluzione dell'aggiunto laminare utilizzando "adjointLaminar" o settando "adjointTurbulence" su "off", al fine di trascurare le equazioni dell'aggiunto relative al calcolo di "ka" e "wa". Nel caso in cui, anche con queste impostazioni, l'aggiunto non dovesse andare a convergenza, significa che il problema risiede nelle condizioni al contorno del caso in esame.

Il primo passo, dopo aver controllato le condizioni al contorno e aver impostato correttamente gli URF, è stato quindi quello di tentare di ricavare una soluzione utilizzando "adjointLaminar". Il calcolo dell'aggiunto a partire dalla soluzione diretta trovata con $k - \omega$ SST ha raggiunto le condizioni di convergenza imposte, come visibile nella pagina successiva in figura 7.5 e 7.6, fornendo dei risultati soddisfacenti. Tuttavia, vi sono alcune considerazioni da fare. Il motivo per il quale è stato implementato il modello di turbolenza per l'aggiunto "adjointkOmegaSST" è che, oltre ad essere ovviamente più preciso di "adjointLaminar" quando si vanno ad estrarre le mappe di sensitività, nel momento in cui si effettua un processo di ottimizzazione su casi industriali, considerando il caso come laminare quando si calcola l'aggiunto, il valore della funzione obiettivo tende a divergere dopo i primi step [21]. Come vedremo questo verrà confermato successivamente anche all'interno del seguente elaborato. Utilizzando invece "adjointkOmegaSST", considerando quindi anche le grandezze aggiunte legate alla turbolenza, si dovrebbero ottenere risultati migliori e la convergenza del valore della funzione obiettivo al crescere degli step di deformazione della geometria.

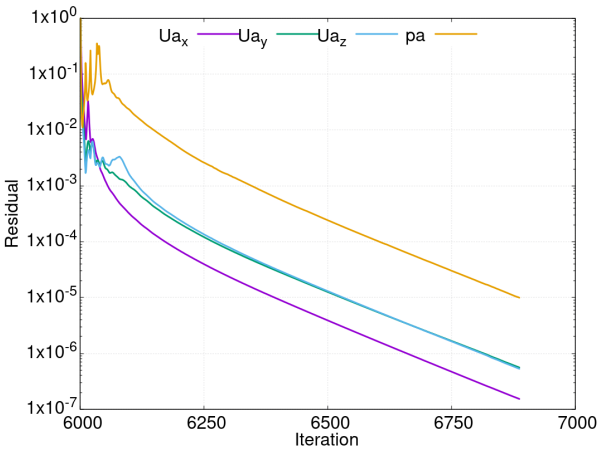


Figura 7.5: Residui calcolo aggiunto adjointLaminar

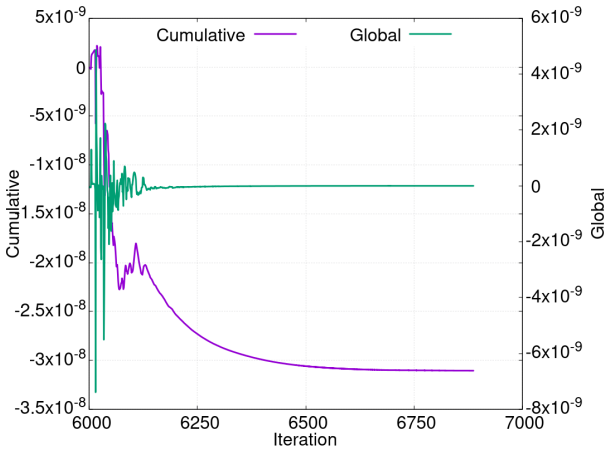


Figura 7.6: Continuità calcolo aggiunto adjointLaminar

7.1.2 Utilizzo di Spalart-Allmaras per il calcolo dell'aggiunto

Come già spiegato nel Capitolo 5, relativo alla discussione dei risultati ottenuti dal calcolo diretto, il modello Spalart-Allmaras non è il più adatto a questo caso specifico, ma ha fornito dei risultati molto più stabili rispetto a quelli ricavati utilizzando il modello di turbolenza $k - \omega$ SST. In figura 7.7 e figura 7.8 è possibile osservare come i residui di tutte le grandezze raggiungano dei valori molto più bassi di quelli visti precedentemente e nelle ultime iterazioni siano caratterizzati da un comportamento meno oscillatorio. In ogni caso anche qui i risultati sono stati mediati sulle ultime 1000 iterazioni per ricercare la massima stabilità possibile del calcolo diretto, in modo da favorire la convergenza dell'aggiunto.

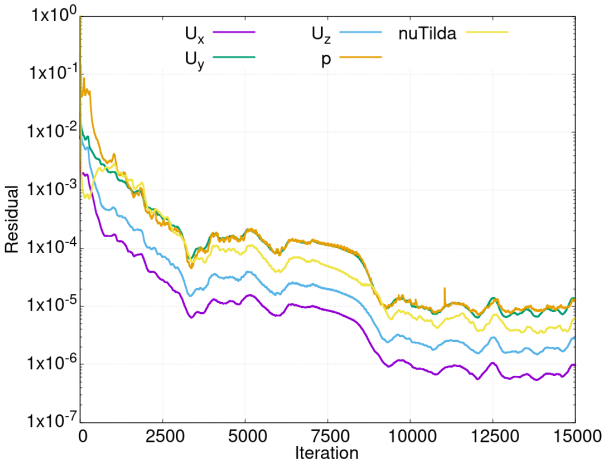


Figura 7.7: Residui calcolo diretto Spalart-Allmaras

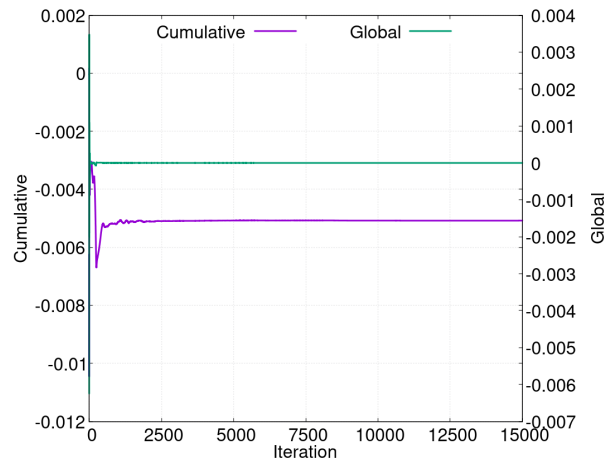


Figura 7.8: Continuità calcolo diretto Spalart-Allmaras

Grazie alla maggiore stabilità del calcolo diretto questa volta, utilizzando delle impostazioni di calcolo molto stabili per quanto riguarda gli schemi di discretizzazione, il calcolo dell'aggiunto ha raggiunto le condizioni di convergenza fissate sui residui mantenendo attivo il modello di turbolenza dell'aggiunto "adjointSpalartAllmaras". Residui e continuità del calcolo sono visibili in figura 7.9 e figura 7.10

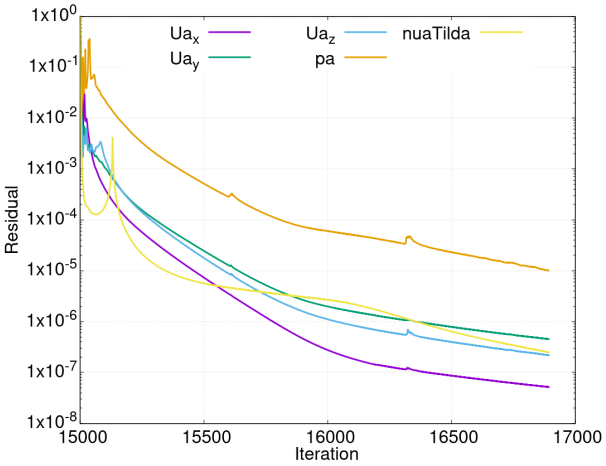


Figura 7.9: Residui calcolo aggiunto adjointSpalartAllmaras

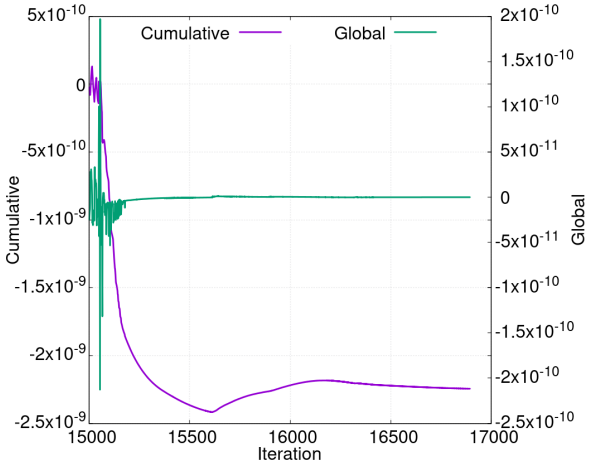


Figura 7.10: Continuità calcolo diretto adjointSpalartAllmaras

Il passo successivo è stato quello di ricavare le mappe di sensitività rispetto a diverse funzioni obiettivo prestabilite e confrontare tra loro i risultati ottenuti per capire come impostare e svolgere al meglio il processo di ottimizzazione.

7.2 Mappe di sensitività

Una volta ottenuta una convergenza dell'aggiunto si è passati al calcolo delle mappe di sensitività. Si è pensato di utilizzare tre funzioni obiettivo distinte per il caso in esame:

- $J = C_l$
- $J = C_l + 5C_d$
- $J = C_l + 5(C_d - C_{d_{Target}})$

La prima si propone di minimizzare il C_l dell'ala che ricordiamo avere valore negativo, in questo modo si va a massimizzare la "downforce" prodotto dall'ala. La seconda è stata utilizzata con l'idea di massimizzare l'efficienza aerodinamica, tuttavia, l'efficienza aerodinamica "E" di un'ala si definisce come $E = \frac{C_l}{C_d}$, per utilizzare una formulazione di questo tipo in OpenFOAM si sarebbe dovuta implementare nel codice una nuova funzione obiettivo, processo lungo e complicato. Si è scelto quindi di utilizzare una funzione obiettivo con lo stesso scopo ma formulata diversamente. Il coefficiente "C_d" è moltiplicato per 5 per far si che all'interno della funzione obiettivo abbia un valore confrontabile con quello iniziale di "C_l", in questo modo vengono considerati con pari importanza durante il calcolo. La terza funzione obiettivo ha uno scopo simile alla seconda, cerca di minimizzare il valore di "C_l" mantenendo il valore di "C_d" costante durante i vari passi dell'ottimizzazione. Di questa funzione non saranno presenti mappe di sensitività in quanto, a causa della struttura stessa della funzione, il secondo termine si annulla nel primo ciclo di calcolo in quanto $C_d = C_{d_{target}}$. Una funzione di questo tipo produce dei risultati diversi da quelli trovati per $J = C_l$ solo nel momento in cui si effettuano più step di ottimizzazione, l'aumento di C_l provoca infatti un aumento del C_d facendo si che il secondo termine non sia più pari a 0 e partecipi al valore di J.

Di seguito vengono mostrate le mappe di sensitività ricavate utilizzando le due funzioni obiettivo $J = C_l$ (Figura 7.16 e Figura 7.17) e $J = C_l + 5C_d$ (Figura 7.18 e Figura 7.19). L'aggiunto è stato calcolato a partire da due risultati del calcolo differenti, in uno è stato utilizzato il modello di turbolenza $k - \omega$ SST, nell'altro Spalart-Allmaras.

L'aggiunto calcolato a partire da $k - \omega$ SST è stato risolto utilizzando all'interno del file "adjointRASProperties" l'impostazione "adjointLaminar". Invece, per l'aggiunto calcolato a partire dai risultati del calcolo diretto ricavati utilizzando Spalart-Allmaras, è stato attivato il modello di turbolenza per l'aggiunto "adjointSpalartAllmaras".

All'interno delle mappe di sensitività non vi sono legende, le rappresentazioni servono unicamente come indicazione della deformazione che ci si aspetta, quando verrà ottimizzata la geometria rispetto alla funzione obiettivo imposta. Le zone in rosso della geometria indicano uno spostamento verso l'esterno (dalla geometria verso il fluido), viceversa quelle blu indicano uno spostamento verso l'interno (dal fluido verso la geometria).

7.2.1 Mappe di sensitività, $J = C_l$

Si osserva come le due mappe di sensitività ottenute con i due diversi modelli di turbolenza siano molto simili tra loro. In entrambi i casi si osserva una parte rossa in corrispondenza del bordo di uscita di ”lower flap” e ”upper flap”, come a voler cercare di aumentare l’incidenza dei flap oppure per generare una struttura simile a quella di un ”gurney flap”, un dispositivo posto nella parte finale dei profili alari per aumentare la portanza di questi aggiungendo una componente di resistenza all’avanzamento molto piccola, quindi largamente utilizzati in ambito automobilistico. Sul ”main flap” nella zona vicina al pilone di sostegno si nota una zona nella quale si cerca di spostare verso il basso l’intradosso del profilo. Questo, oltre ad aumentarne la curvatura, avvicina l’ala al terreno, aumentando l’effetto suolo e incrementando la ”downforce” complessiva.

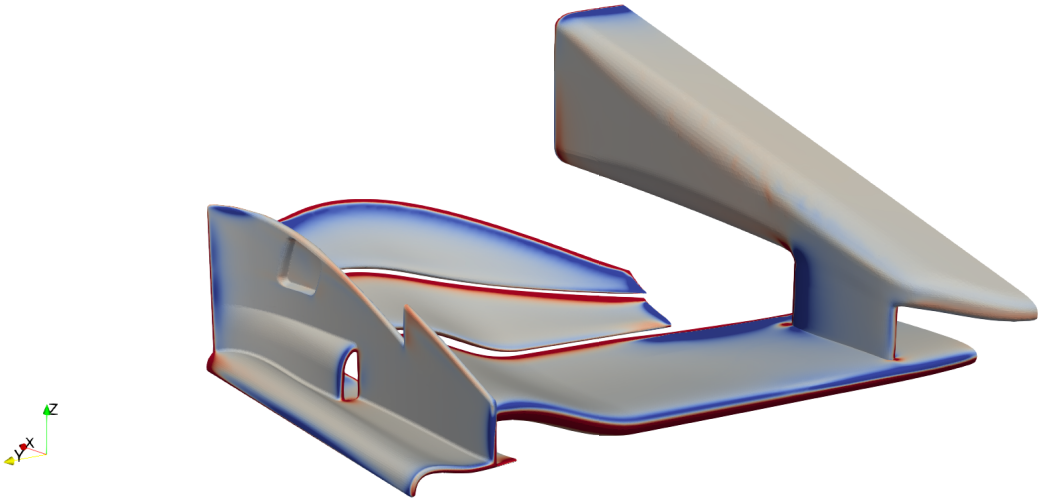


Figura 7.11: Mappa di sensitività $J = C_l$ - adjointLaminar

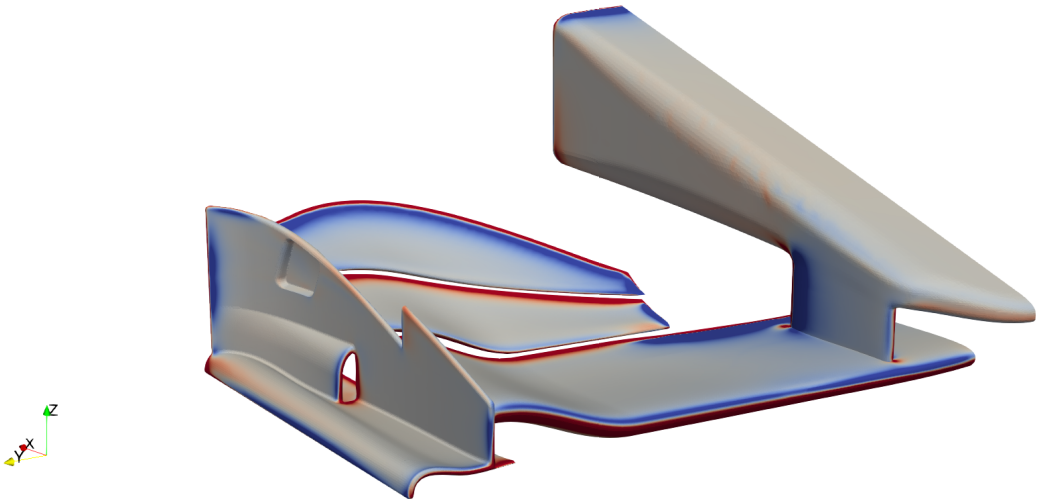


Figura 7.12: Mappa di sensitività $J = C_l$ - adjointSpalartAllmaras

7.2.2 Mappe di sensitività, $J = C_l + 5C_d$

Anche in questo caso le due mappe di sensitività sono molto simili tra loro. Contrariamente al caso precedente qui si cerca di aumentare la deportanza generata dall'ala e, contemporaneamente, ridurre la resistenza all'avanzamento della stessa. Si osserva quindi una mappa di sensitività dove si cerca di assottigliare i profili, dandogli una forma più allungata in corrispondenza del bordo di attacco. Questo, si nota in particolare sul naso dell'ala, si osserva una zona rossa circondata da una leggera sfumatura blu, significa che si cerca di allungare il "nose" e assottigliarlo per fornirgli una forma più acuminata. L'"upper flap", inoltre, è caratterizzato da una colorazione rossa su buona parte del suo intradosso, questo indica un sollevamento del profilo e una diminuzione della deflessione complessiva data al fluido dall'ala.

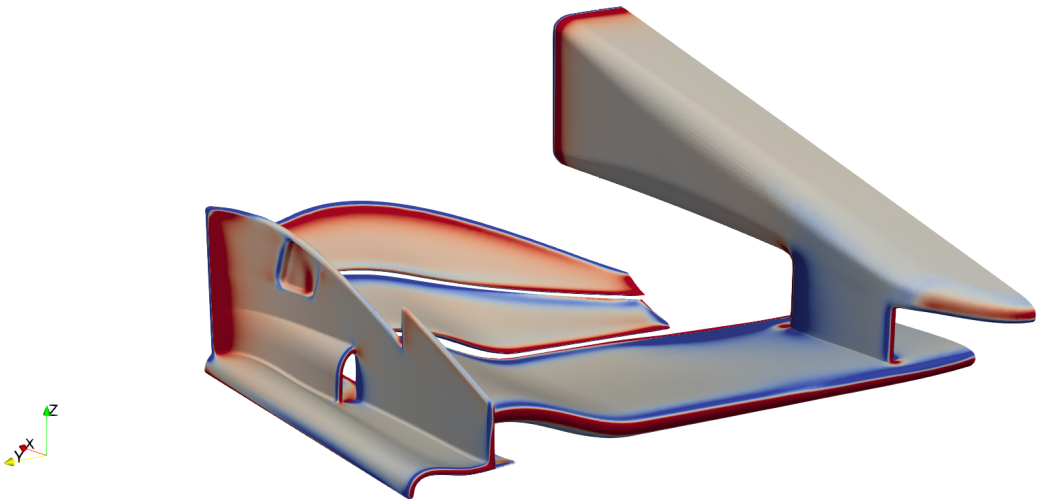


Figura 7.13: Mappa di sensitività - $J = C_l + 5C_d$ - adjointLaminar

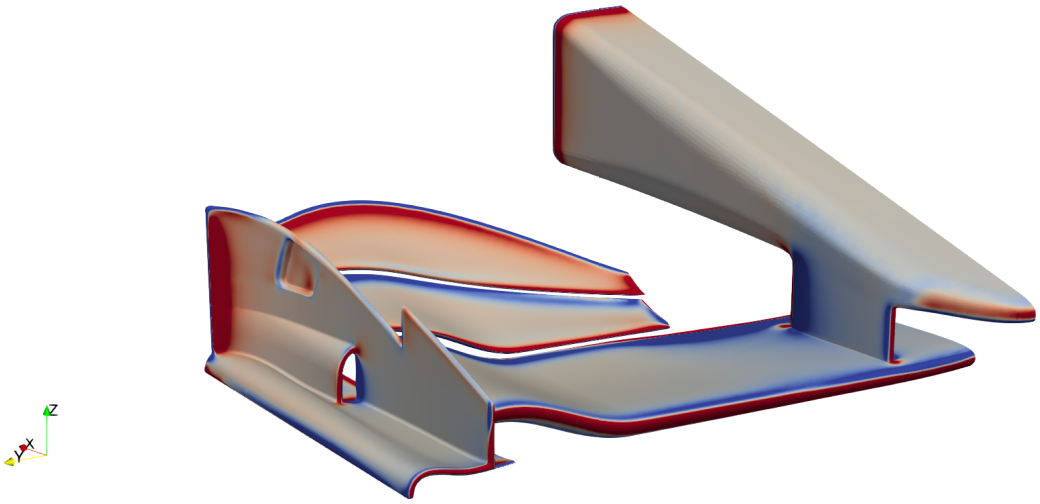


Figura 7.14: Mappa di sensitività - $J = C_l + 5C_d$ - adjointSpalartAllmaras

A questo punto vengono mostrate nel dettaglio la sensitività e, attraverso dei vettori, la direzione della deformazione che ci si aspetta di ricavare su alcune sezioni dell'ala. Nelle immagini saranno mostrati i confronti tra le varie mappe di sensitività su una sola sezione, quella in $y/s = 0.53$ mostrata in figura 7.15. Questo perché a causa della presenza dei vettori le immagini ricavate sulle altre sezioni risultavano poco chiare e confusionarie. Quanto detto a proposito della sezione in esame si applica comunque anche a tutte le altre sezioni, non particolarmente diverse da quella esposta, e all'ala in generale.

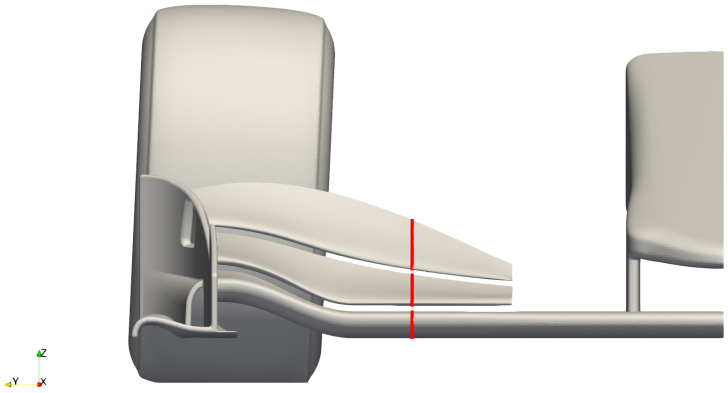


Figura 7.15: Sezione dell'ala utilizzata per l'analisi della sensitività

Anche all'interno delle immagini nelle pagine successive non è presente la legenda, si applica lo stesso discorso fatto per le mappe di sensitività nei sottocapitoli precedenti. Questa volta, però, saranno presenti dei vettori rappresentati da frecce colorate, queste, indicano direzione e verso della deformazione della geometria data dalla mappa di sensitività. L'intensità del colore delle frecce indica l'entità della deformazione della geometria in un punto specifico.

A riconferma di quanto detto precedentemente le figure ottenute con "adjointLaminar" e quelle ottenute con "adjointSpalartAllmaras" sono molto simili tra loro, quasi uguali. Per quanto riguarda le prime due figure (Figura 7.16 e Figura 7.17), relative alla funzione obiettivo $J = C_l$, si osserva la tendenza a deformare il "main flap" per cercare di aumentarne la curvatura e avvicinare l'estradosso del profilo al terreno, per aumentare l'effetto suolo. Per quanto riguarda "lower flap" e "upper flap", invece, si osserva come si cerchi di spostare il bordo di attacco verso il basso e il bordo di uscita verso l'alto. In questo modo si aumenta l'incidenza dei profili e quindi la deportanza prodotta dall'ala.
Per quanto riguarda le figure relative alla funzione obiettivo $J = C_l + 5C_d$ (Figura 7.18 e Figura 7.19) è possibile osservare come le frecce indichino la tendenza ad assottigliare i profili, soprattutto in corrispondenza del bordo di attacco del "main flap". Per quanto riguarda il resto della geometria si cerca di lavorare sull'inarcamento dell'estradosso nei punti in vicinanza del "leading edge". Questo viene fatto per aumentare la velocità del flusso e ridurre la pressione nella parte bassa dei profili, generando più deportanza. Inoltre, come già accennato precedentemente, per cercare di contenere il drag c'è una tendenza a spostare i profili verso l'alto in modo da ridurre la deflessione totale del flusso, questo è particolarmente visibile grazie ai vettori su "lower flap" e "upper flap".

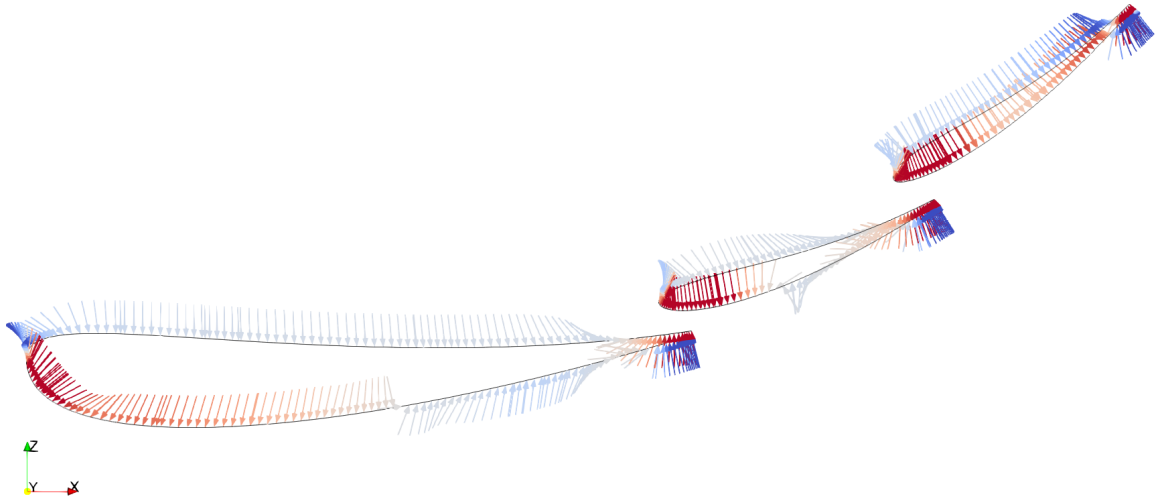


Figura 7.16: Sensitività e direzione di deformazione - $J = C_l$ - adjointLaminar

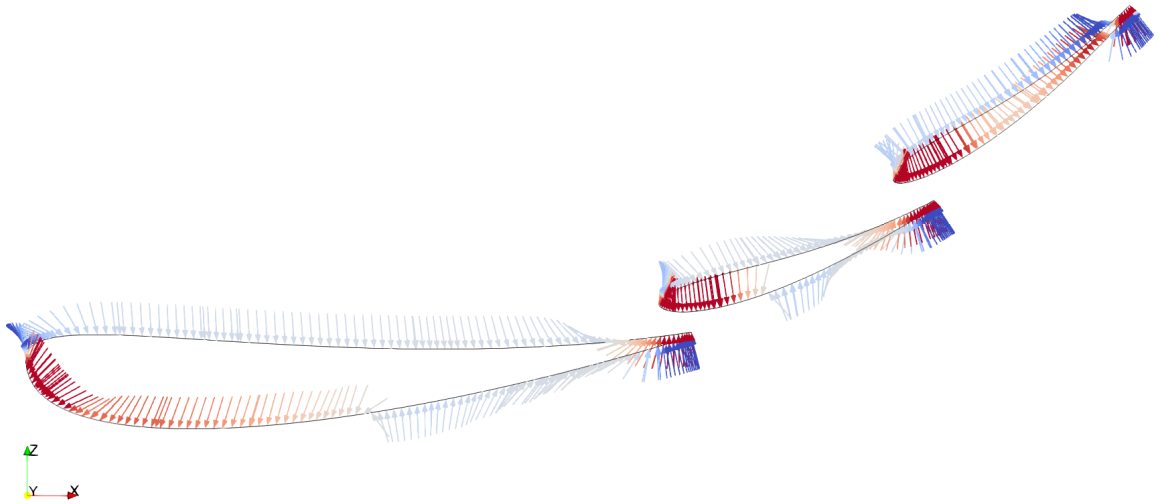


Figura 7.17: Sensitività e direzione di deformazione - $J = C_l$ - adjointSpalartAllmaras

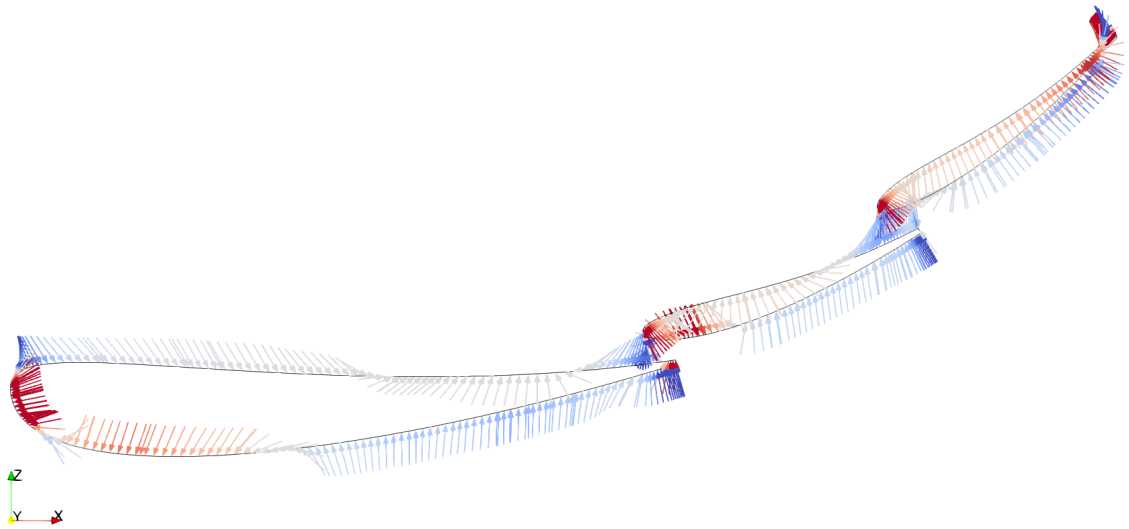


Figura 7.18: Sensitività e direzione di deformazione - $J = C_l + 5C_d$ - adjointLaminar

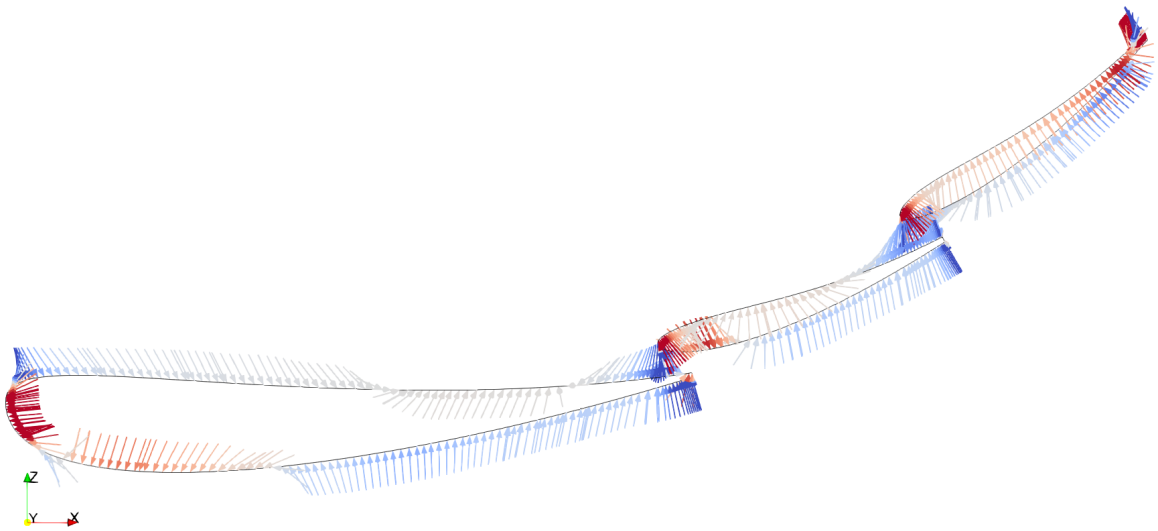


Figura 7.19: Sensitività e direzione di deformazione - $J = C_l + 5C_d$ - adjointSpalartAllmaras

7.3 Processo di ottimizzazione e deformazione della geometria

Trovate le mappe di sensitività è stata definita una prima box di punti per effettuare la prima deformazione della geometria. Questo processo viene fatto inserendo nella directory ”constant” il file ”dynamicMeshDict” al cui interno troviamo tutte le informazioni relative alla box di deformazione e ai punti da essa contenuti. Una volta definite la posizione e la dimensione della box è stato necessario decidere quanti punti di controllo inserire. Essendo la potenza di calcolo a disposizione abbastanza limitata è stato utilizzato un numero di punti ridotto, concentrandosi sull'avere più controllo nella direzione ”z”, quella che dalle mappe di sensitività è sembrata essere la più importante da controllare sotto il punto di vista della deformazione della geometria. Quindi sono stati utilizzati 4 punti in direzione ”x”, 4 punti in direzione ”y” e 8 punti in direzione ”z”, come visibile in figura 7.20 e 7.21.

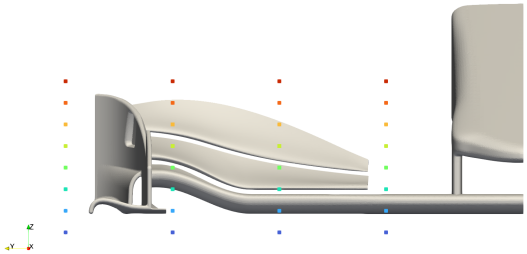


Figura 7.20: Box di deformazione visuale frontale

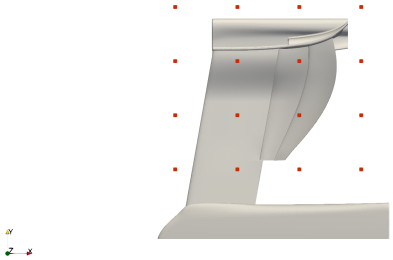


Figura 7.21: Box di deformazione visuale dall'alto

La deformazione è avvenuta utilizzando una tecnica denominata ”Free Form Deformation” (FFD). Una volta posizionati i punti di controllo viene effettuata attraverso un algoritmo prestabilito, nel nostro caso ”b-spline”, la mappatura di tutti i punti della griglia di calcolo rispetto ai punti di controllo di cui sopra. Attraverso i dati ricavati dalle mappe di sensitività viene poi deformata la mesh secondo dei polinomi, in questo caso scelti di grado 3, meno dettagliati ma con la quasi certezza di ottenere una forma finale con una buona continuità.

Per un primo tentativo di deformazione non sono stati imposti limiti sul massimo spostamento dei punti della geometria, è servita infatti solo per avere una prima idea di quelle che potevano essere le deformazioni ottenute a partire dalle varie mappe di sensitività, ricavate con la risoluzione dell'aggiunto. Abbiamo quindi prodotto dei confronti tra i risultati e le deformazioni ottenute utilizzando sia ”adjointLaminar” che ”adjointSpalartAllmaras” e le due funzioni obiettivo $J = C_l$ e $J = C_l + 5C_d$. Queste, sono rispettivamente visibili in figura 7.22 e figura 7.23 alla pagina successiva. In rosso è visibile la deformazione ottenuta dalla risoluzione dell'aggiunto con ”adjointSpalartAllmaras”, in blu quella ottenuta con ”adjointLaminar”.

Si osserva come le variazioni della geometria siano coerenti con quanto detto nell'analisi delle mappe di sensitività e, soprattutto, si nota come le deformazioni ottenute con ”adjointSpalartAllmaras” e ”adjointLaminar”, a partire rispettivamente dai calcoli diretti eseguiti utilizzando il modello di turbolenza Spalart-Allmaras e $k - \omega$ SST, sono praticamente identici.

Questo è il motivo principale per cui per il caso in esame si è deciso di effettuare l'ottimizzazione utilizzando sia il modello di turbolenza $k - \omega$ SST che Spalart-Allmaras per la risoluzione del calcolo diretto, nonostante quest'ultimo non fosse il miglior modello di turbolenza da utilizzare per una geometria come quella analizzata.

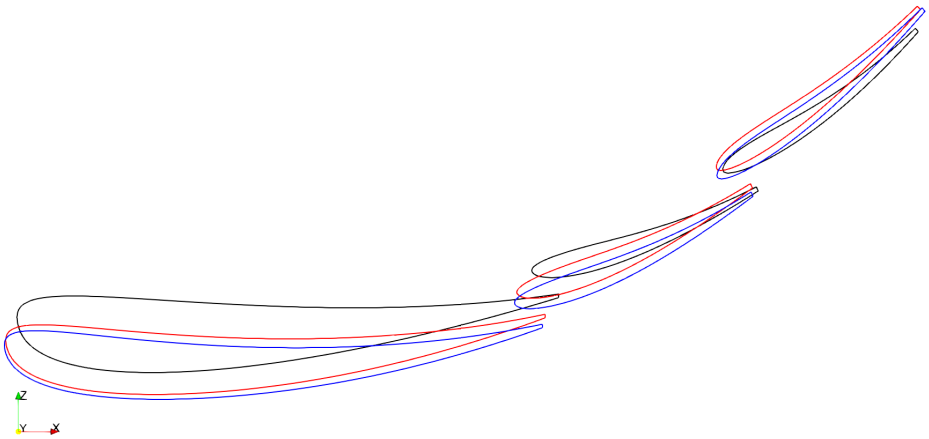


Figura 7.22: Deformazione iniziale - $J = C_l$ - adjointSpalartAllmaras, adjointLaminar

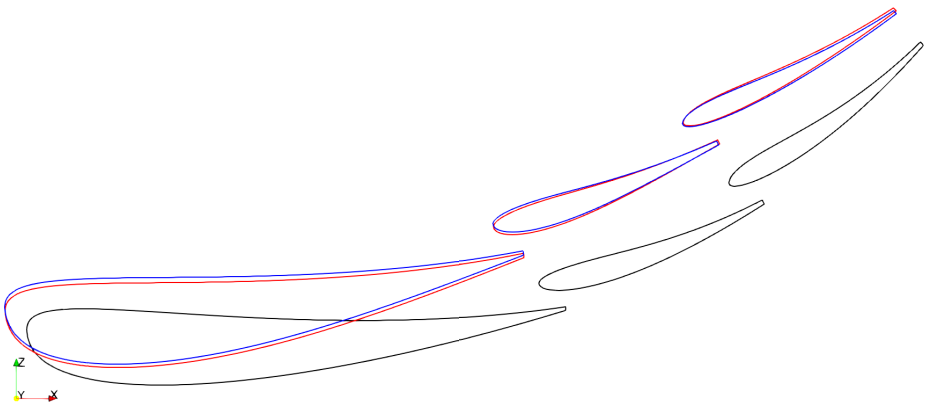


Figura 7.23: Deformazione iniziale - $J = C_l + 5C_d$ - adjointSpalartAllmaras, adjointLaminar

7.4 Ottimizzazione di forma finale

Per l'ottimizzazione di forma finale è stato scelto di fare 10 passi di deformazione, per ognuno è necessario effettuare calcolo diretto, risoluzione dell'aggiunto e deformazione della griglia di calcolo. Fare questo tipo di processo per 10 volte è molto dispendioso in termini di tempo, quindi, quando possibile, sono state utilizzate le macchine presenti in Dallara che permettono di svolgere il calcolo in parallelo su un massimo di 256 core, mentre con il computer normalmente utilizzato presso l'università si aveva disponibilità di un massimo di 16 core. Idealmente si sarebbero dovuti impostare molti più step di deformazione in modo da portare a convergenza il valore della funzione obiettivo. Questo, non è stato possibile principalmente per due motivi: il primo è legato alle tempistiche come appena spiegato, il secondo è legato al fatto che non viene fatto un "remeshing" dopo ogni deformazione, quindi gli elementi della mesh vengono deformati insieme alla geometria. Nel momento in cui si arriva a forti deformazioni si rischia di incorrere in problemi quali celle con volume negativo o valori molto elevati di "skewness" o "aspect-ratio". Questo è quello che è successo durante il processo di ottimizzazione con "adjointSpalartAllmaras" che è stato appunto fermato allo step numero 9 a causa di problemi su svariate celle della mesh.

Il file "optimisationDict" nella directory "system" è stato modificato inserendo nel dictionary "optimisationManager" la funzione "steadyOptimisation". A questo punto "startTime" ed "endTime" in "controlDict" non sono più istante iniziale e finale del singolo calcolo ma rappresentano step iniziale e finale della deformazione. Inoltre altre modifiche sono state apportate sempre in "optimisationDict" nel dictionary "optimisation" che assumerà una forma del tipo:

```
optimisation
{
    designVariables
    {
        type                    shape;
        shapeType               volumetricBSplines;
        sensitivityType         shapeESI;
        smoothSensitivities     true;
        iters                   5000;
        patches                 ("wing-patches");
        maxInitChange           1.e-3;
    }
    updateMethod
    {
        method conjugateGradient;
    }
}
```

Da notare il tipo si sensitività "shapeESI" specifica per effettuare variazioni di forma della geometria, rispetto alla tipologia "surface", utilizzata per le mappe di sensitività, presenta dei termini in più utili ad aumentarne la precisione. Il termine "maxInitChange" gestisce il massimo passo iniziale di deformazione, è stato lasciato volutamente molto piccolo per evitare forti oscillazioni nell'andamento del valore della funzione obiettivo, che si possono presentare nel caso in cui lo step di deformazione fosse troppo grande.

Infine, come ”updateMethod” è stato scelto ”conjugateGradient” che è un giusto compromesso tra robustezza e precisione del risultato, soprattutto per casi industriali e caratterizzati da geometrie complicate come quella in esame.

Per motivi principalmente legati alle tempistiche è stato deciso di effettuare due ottimizzazioni di forma utilizzando solo la funzione obiettivo $J = C_l$.

In figura 7.24 e figura 7.25 è possibile osservare i due grafici che mostrano l'andamento di ”C_l” durante i passi di ottimizzazione, rispettivamente per i due casi in cui sono stati utilizzati ”adjointLaminar” e ”adjointOptimisationFoam”.

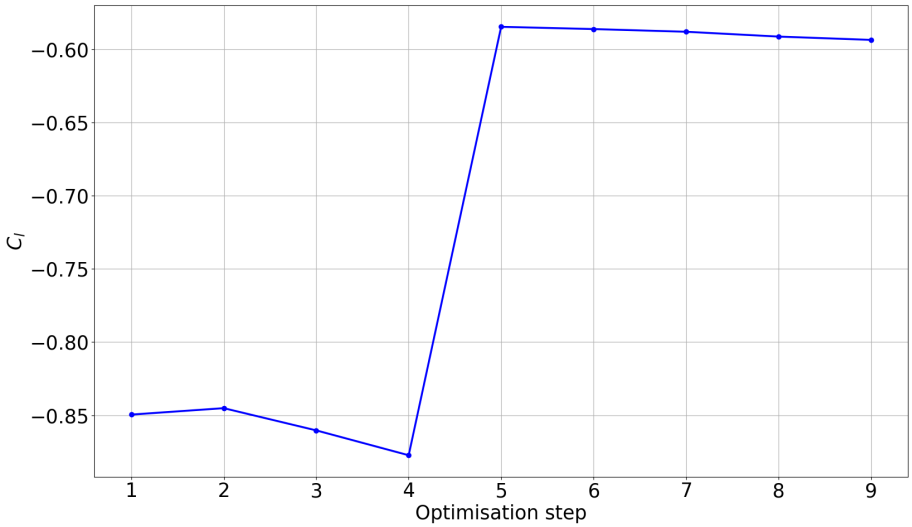


Figura 7.24: C_l vs Optimisation step - ”adjointLaminar”

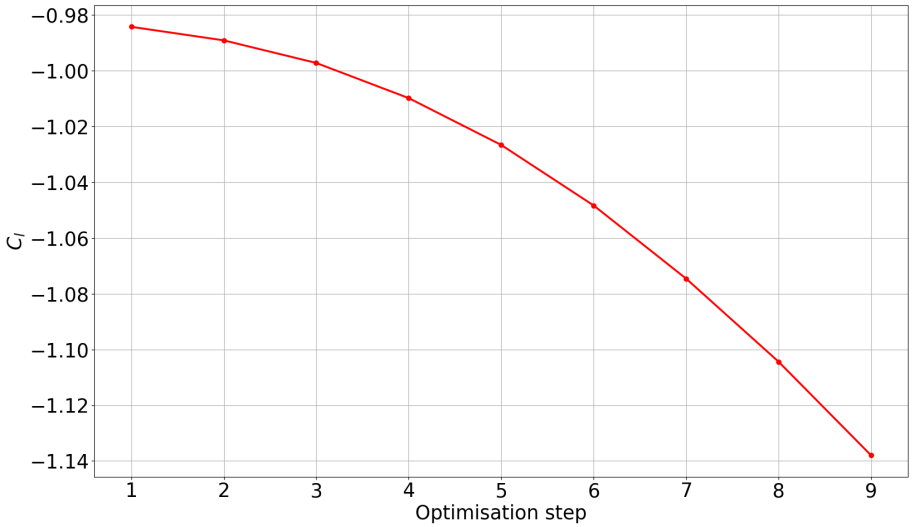


Figura 7.25: C_l vs Optimisation step - ”AdjointSpalartAllmaras”

Si osserva quanto già accennato all'interno del capitolo riguardo la risoluzione dei problemi. Utilizzando l'ipotesi secondo cui il caso è di tipo "frozen turbulence" e utilizzando l'impostazione "adjointLaminar" si hanno delle imprecisioni nella mappa di sensitività che portano alla divergenza del valore della funzione obiettivo. Invece, utilizzando il modello di turbolenza "adjointSpalartAllmaras", si osserva come il valore della funzione obiettivo continui a diminuire nei vari step di ottimizzazione.

Nel secondo caso il valore di C_l è passato da -0.984 a -1.137, la deportanza generata dall'ala è quindi aumentata del 15,57%. Non essendoci alcun controllo sul "drag" generato dall'ala è aumentato anch'esso, il C_d è infatti passato da 0.188 a 0.216 con un aumento del 14,75%.

Nelle pagine successive verranno mostrate le variazioni di geometria sulle sezioni di controllo utilizzate fino ad ora, ad ogni sezione è associato un grafico del "C_p" dove ne viene confrontata la distribuzione sui profili prima e dopo la deformazione della geometria. A seguire verranno inoltre mostrati dei confronti su alcune delle sezioni di controllo tra contour di velocità, coefficiente di pressione (C_p) e coefficiente di pressione totale (C_{pT}) prima e dopo l'ottimizzazione di forma.

7.4.1 Confronto tra le distribuzioni di C_p prima e dopo la deformazione

Come appena spiegato nelle pagine seguenti dalla figura 7.26 alla figura 7.31 è possibile osservare i risultati ottenuti dal processo di ottimizzazione di forma finale, eseguito utilizzando come modello di turbolenza per il calcolo diretto "Spalart-Allmaras" e mantenendo il modello di turbolenza per l'aggiunto "adjointSpalartAllmaras" attivo. Sono stati effettuati 9 step di deformazione, all'interno delle immagini viene preso in considerazione il confronto tra la geometria originale (rappresentata in tutte le immagini dal colore blu) e quella raggiunta all'ultimo step di deformazione (rappresentata dal colore rosso). Per semplificare la comprensione ogni pagina contiene una sola immagine composta di tre sotto immagini, dall'alto verso il basso si incontrano:

- Rappresentazione sulla geometria originale della sezione analizzata.
- Confronto tra geometria dei profili originali (blu) e modificati (rosso)
- Confronto tra le distribuzioni del C_p sui profili originali (blu) e modificati (rosso)

Esiste un legame tra la distribuzione di C_p su un profilo e il "lift" generato dallo stesso che può essere espresso come:

$$C_l = \int_0^1 (C_{p,p} - C_{p,s})\, d(\frac{x}{c}) \tag{7.1}$$

Dove "$C_{p,p}$" è la distribuzione del coefficiente di pressione sul "Pressure side", quindi nel nostro caso sulla parte superiore del profilo, mentre "$C_{p,s}$" è quella sulla parte inferiore, il "suction side". Riassumendo, è possibile dire che l'area compresa all'interno della distribuzione di C_p su un profilo rappresenta il "lift" generato dallo stesso (per un valore di apertura alare unitario). Quindi, nel nostro caso in particolare, più area è compresa all'interno delle curve, maggiore sarà la "downforce" generata dall'ala su quella particolare sezione.

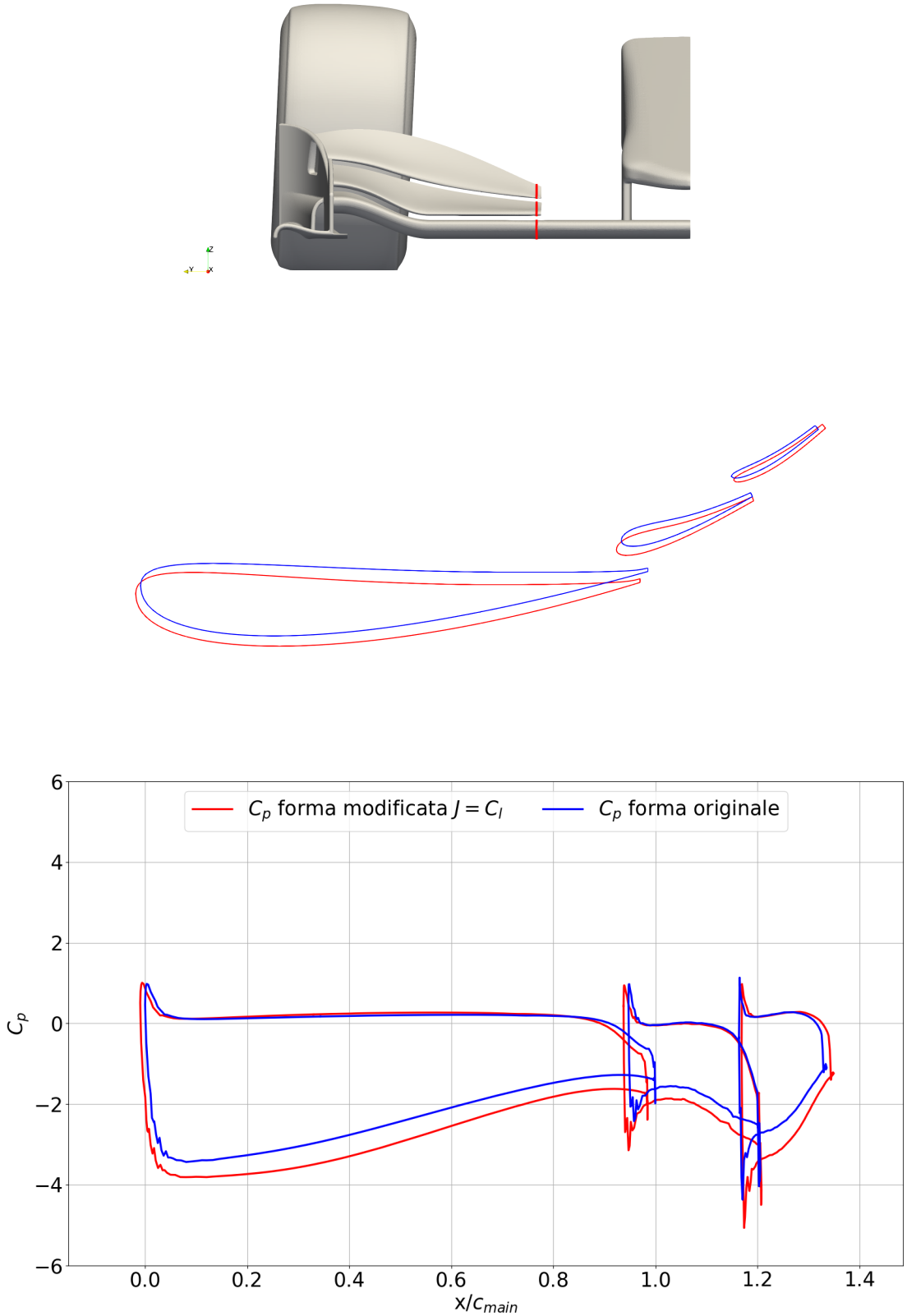
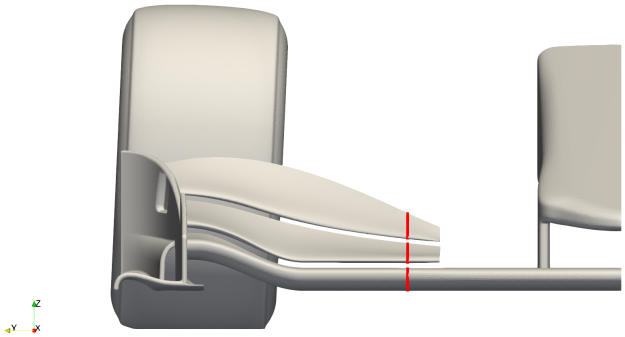
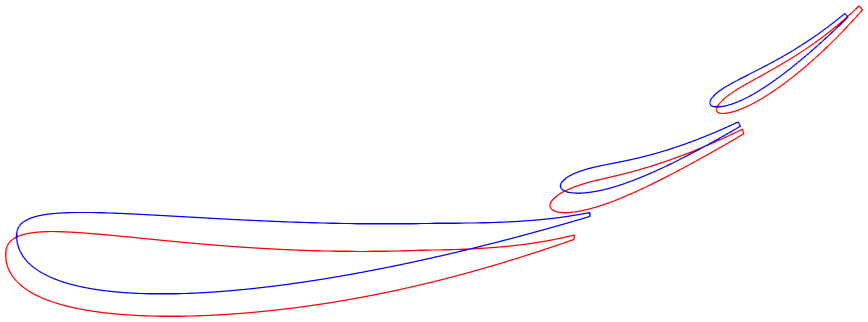


Figura 7.26: Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.37$





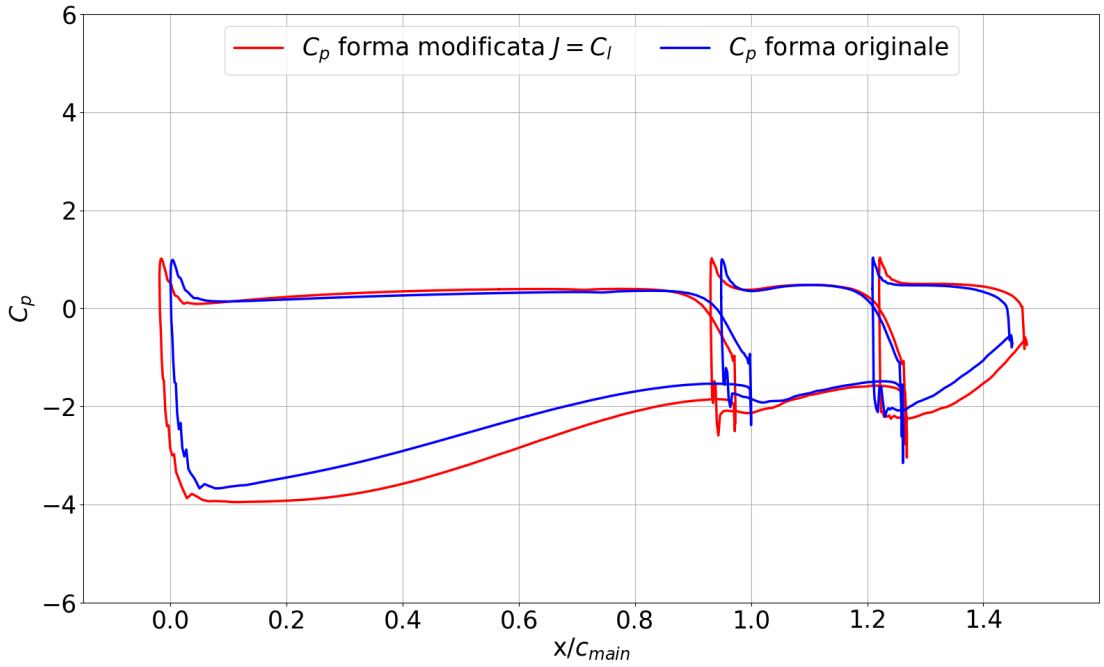


Figura 7.27: Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.42$

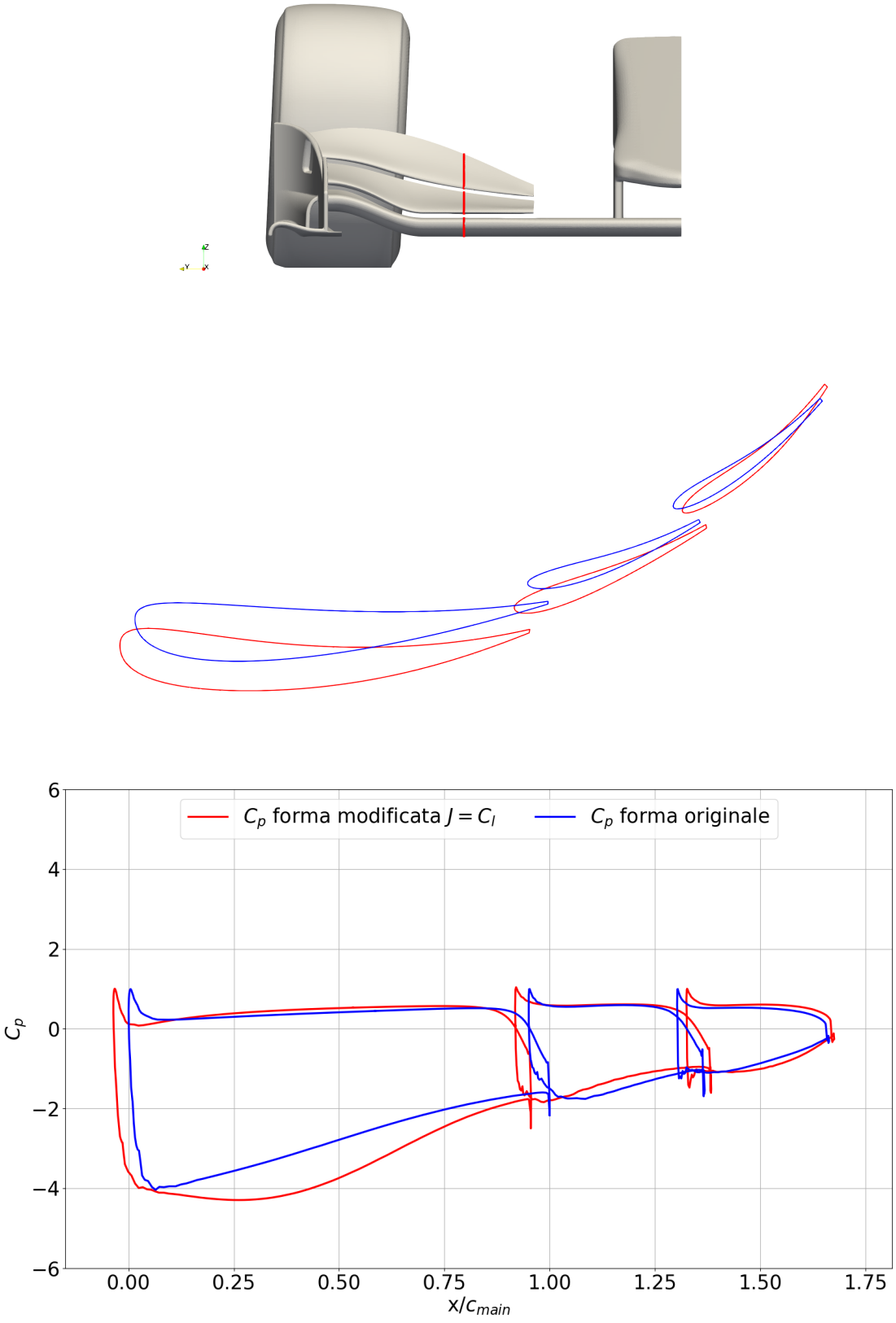


Figura 7.28: Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.53$

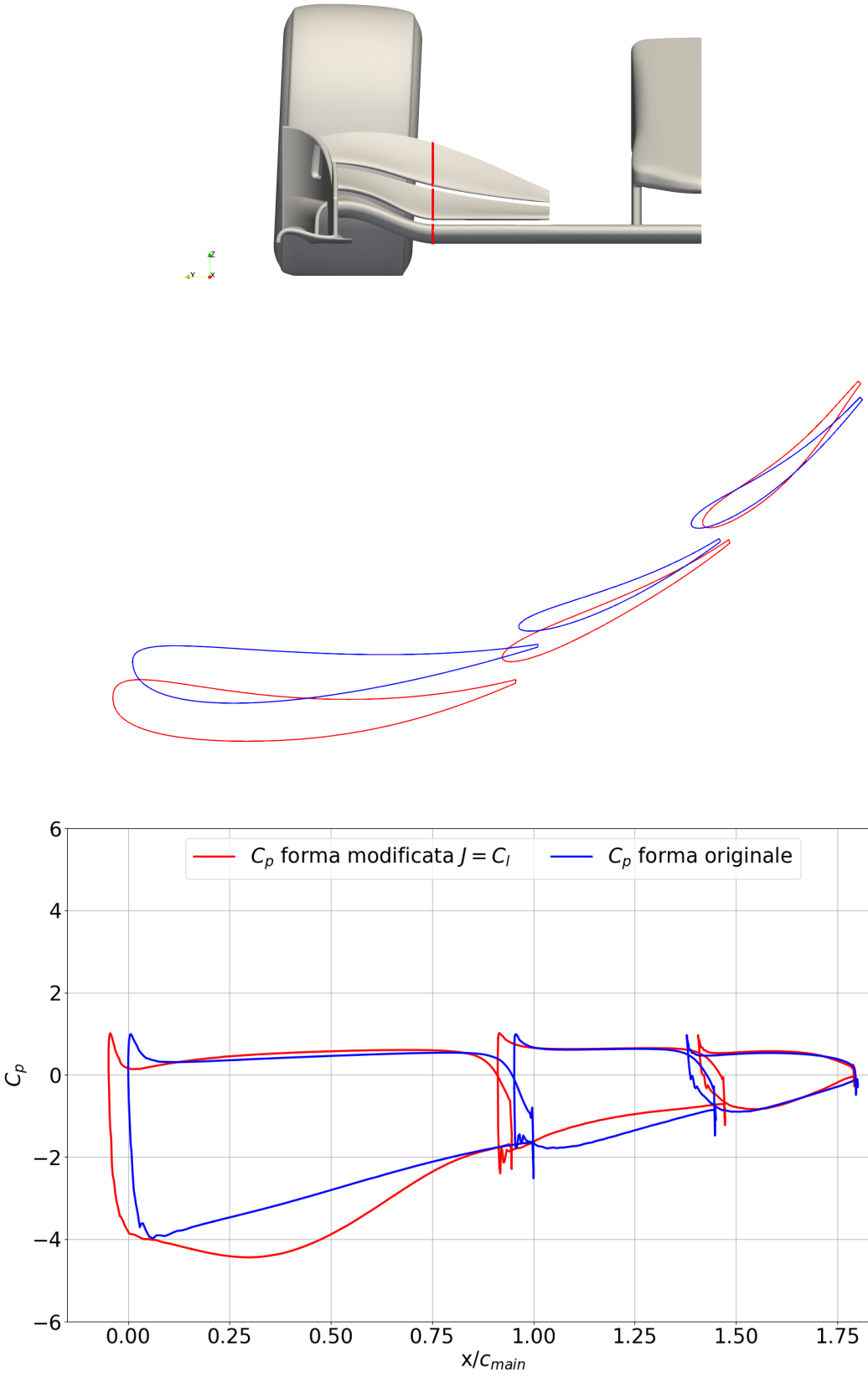


Figura 7.29: Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.63$

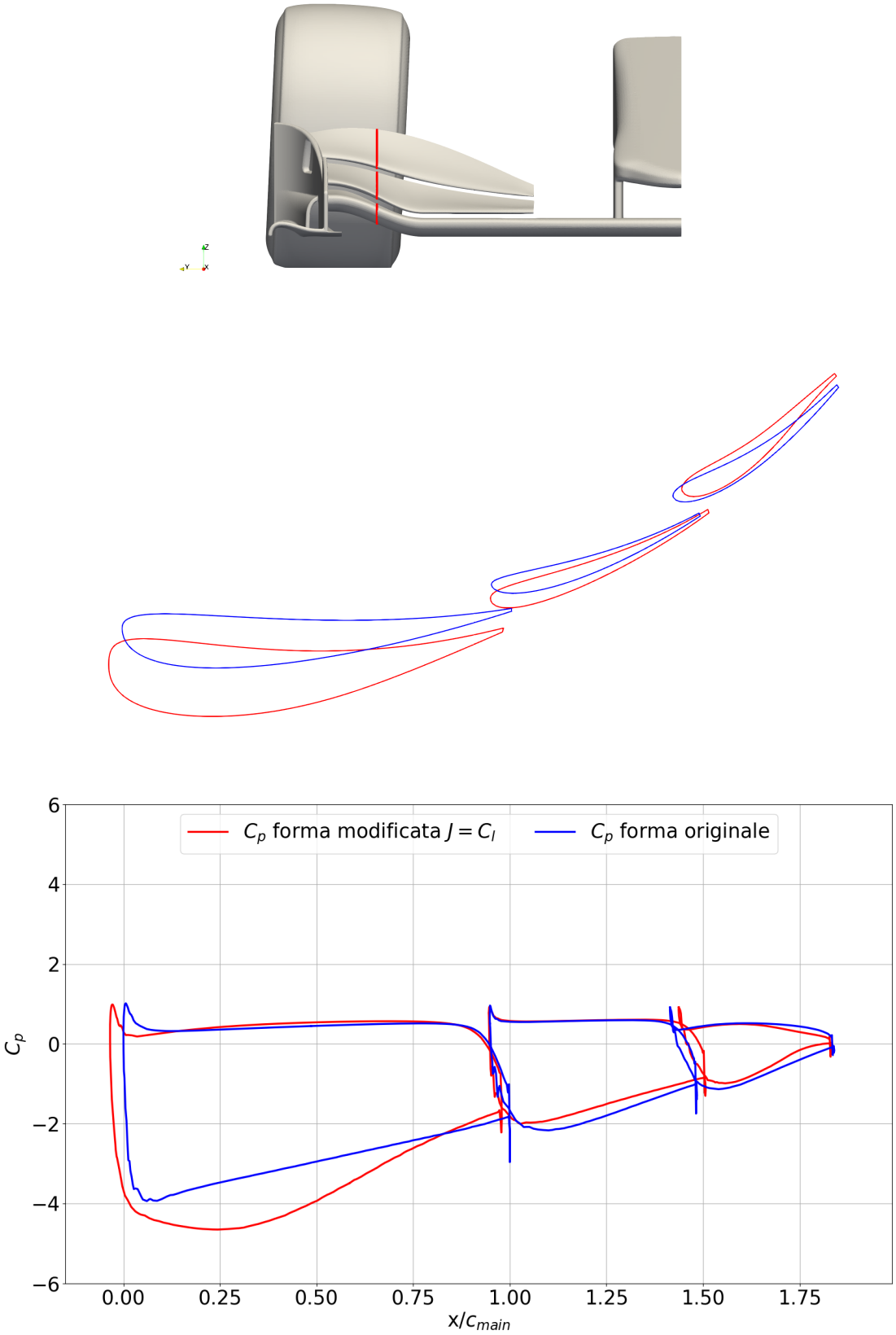


Figura 7.30: Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.74$

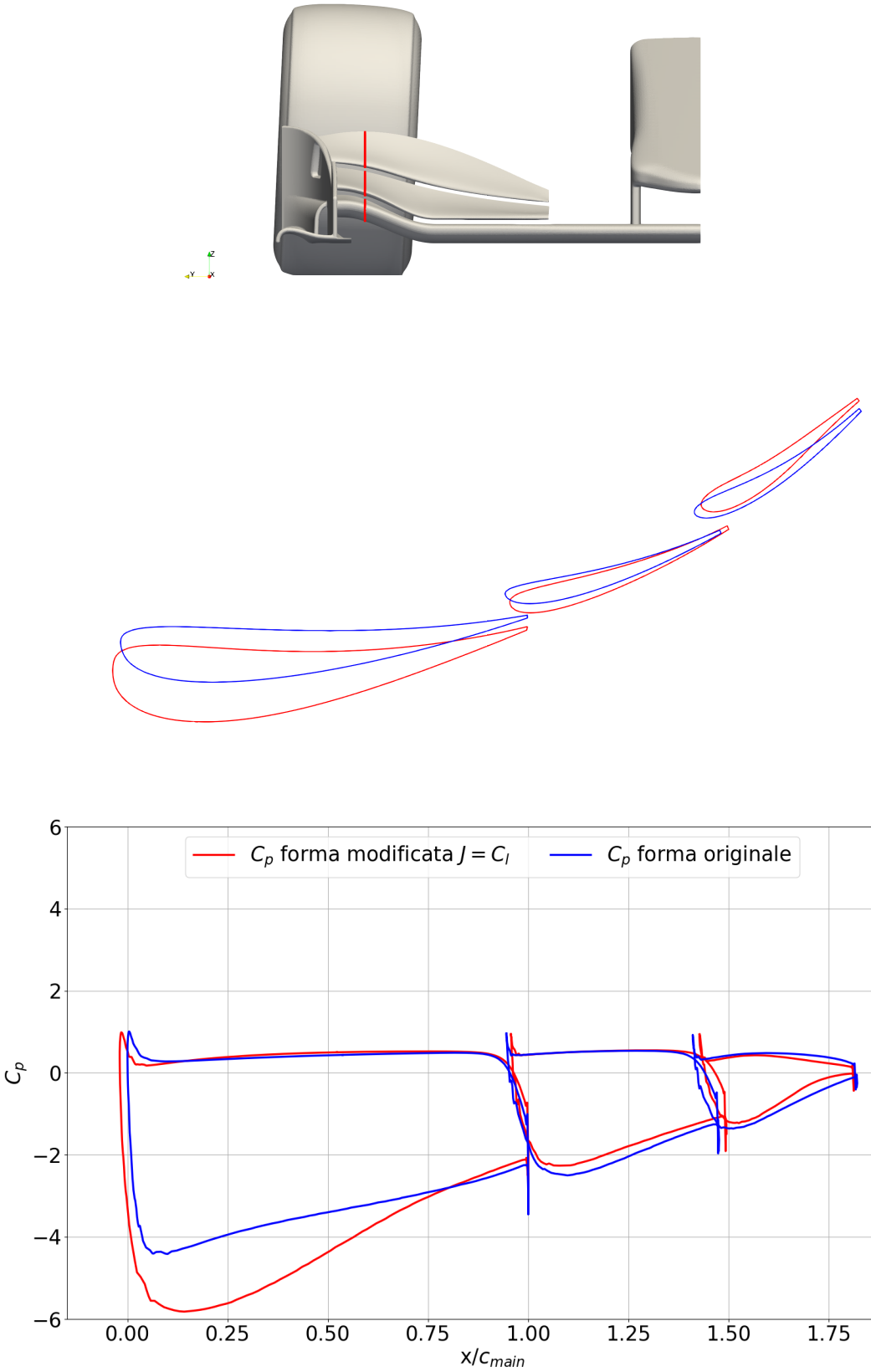


Figura 7.31: Sezione, deformazione della geometria e distribuzione di C_p - $y/s = 0.79$

Analizzando i risultati ottenuti è possibile riscontrare come le previsioni fatte quando sono state commentate le mappe di sensitività fossero corrette. Su tutte le sezioni si osserva la tendenza ad aumentare curvatura e vicinanza al suolo del profilo relativo al "main flap". Per quanto riguarda "lower flap" e "upper flap" si nota sempre un aumento della curvatura, unito questa volta ad un aumento dell'incidenza dei profili e della lunghezza della corda in alcuni casi, quest'ultimo tipo di variazione è particolarmente visibile alla figura 7.28 e alla figura 7.29. Inoltre, osservando la deformazione subita dall'"upper flap" in figura 7.29, figura 7.30 e figura 7.31, è possibile notare quanto già precedentemente accennato riguardo la formazione in vicinanza del bordo di uscita di una maggiore curvatura, per cercare di generare una struttura simile a quella di un "gurney flap". Per averne la certezza, bisognerebbe probabilmente utilizzare box di deformazione con un numero molto elevato di punti di controllo in quella zona, in questo modo la deformazione risulterebbe molto più dettagliata.

Per quanto riguarda le distribuzioni del "C_p" si osservano due comportamenti differenti. Dove le incidenze di "lower flap" e "upper flap" sono ridotte la deformazione fa si che si abbia un contributo di tutti i flap sull'aumento di deportanza, principalmente legato alla diminuzione del C_p nella parte inferiore dell'ala, come si osserva in figura 7.26 e figura 7.27.

Sulle sezioni dove "lower flap" e "upper flap" hanno delle incidenze più elevate, quindi da figura 7.28 a figura 7.31, si osserva come gran parte del contributo all'aumento della "downforce" sia dato dal "main flap". Invece, le distribuzioni del coefficiente di pressione su "lower flap" e "upper flap" indicano un mantenimento della "downforce" e in alcuni casi addirittura una leggera perdita rispetto a quella data dai profili originali. Questo, probabilmente è legato alla forte deflessione del flusso in quelle zone che porta verso una condizione di separazione dello strato limite sull'estradosso dei profili. Su queste sezioni, quindi, è interessante osservare quanto sia efficacie l'effetto suolo nel far diminuire il "C_p" sull'estradosso dei profili e il fatto che la deformazione della geometria cerchi proprio di avvicinare l'estradosso dell'ala al terreno per massimizzare questo fenomeno.

7.4.2 Confronto contour di Coefficiente di Pressione e Coefficiente di Pressione Totale

Nelle pagine successive è possibile osservare un confronto, tra geometria originale e geometria modificata, riguardo i contour di Coefficiente di Pressione "C_p" e Coefficiente di Pressione Totale "C_{pT}". Per alleggerire l'analisi verranno commentate solo tre sezioni per ogni caso, le più significative in termini di differenze riscontrate tra caso originale e deformato. Le sezioni analizzate saranno rappresentate sopra ogni coppia di contour, il primo contour è rappresentativo del caso con geometria originale mentre il secondo rappresenta il caso con geometria modificata. Entrambe le figure fanno riferimento alla stessa legenda mostrata ai piedi della seconda figura, in questo modo è possibile apprezzarne al meglio le differenze.

In particolare osservando i contour di "C_p" in figura 7.33 e figura 7.34 si può apprezzare come il coefficiente di pressione, a causa della maggiore vicinanza del "main flap" al terreno, subisca una forte diminuzione rispetto al caso originale.

Per quanto riguarda il coefficiente di pressione totale è in grado di fornire indicazioni importanti a proposito della separazione dello strato limite, di cui si è parlato precedentemente. Si nota infatti dalla figura 7.35 e dalla figura 7.36 come lo strato limite nel caso con geometria modificata subisca una separazione in vicinanza del bordo di uscita del "main Flap", dove il flusso è soggetto ad un gradiente avverso di pressione. La separazione va poi ad influenzare le performance di "lower flap" e "upper flap". In figura 7.37 nonostante le forti curvature dei profili non si osserva lo stesso incremento di separazione da caso originale a deformato. Questo, potrebbe essere legato ai disturbi generati dall'"endplate", compreso anch'esso nella deformazione della geometria, che influenzano il flusso. Si osserva, infatti, sotto i profili, una zona in cui è presente una riduzione del "C_{pT}", legata probabilmente alla presenza di un vortice generato dall'"endplate". Le variazioni del flusso causate da questo potrebbero essere utili ad energizzare lo strato limite in quella zona, prevenendone la separazione. Quest'ultimo aspetto è sicuramente molto interessante come spunto per studi o sviluppi futuri.

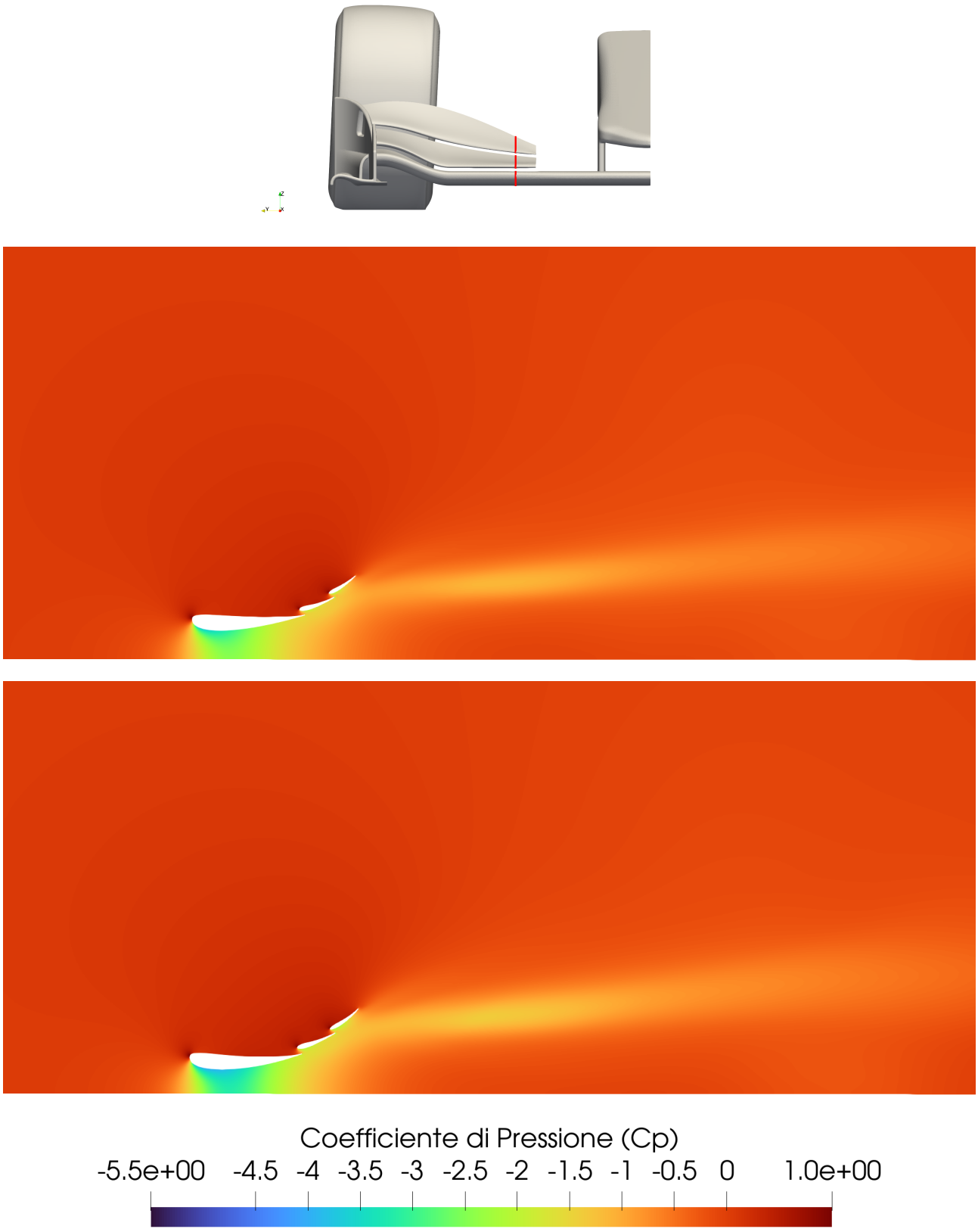


Figura 7.32: Contour di C_p - Geometria originale vs Geometria modificata - $y/s = 0.42$

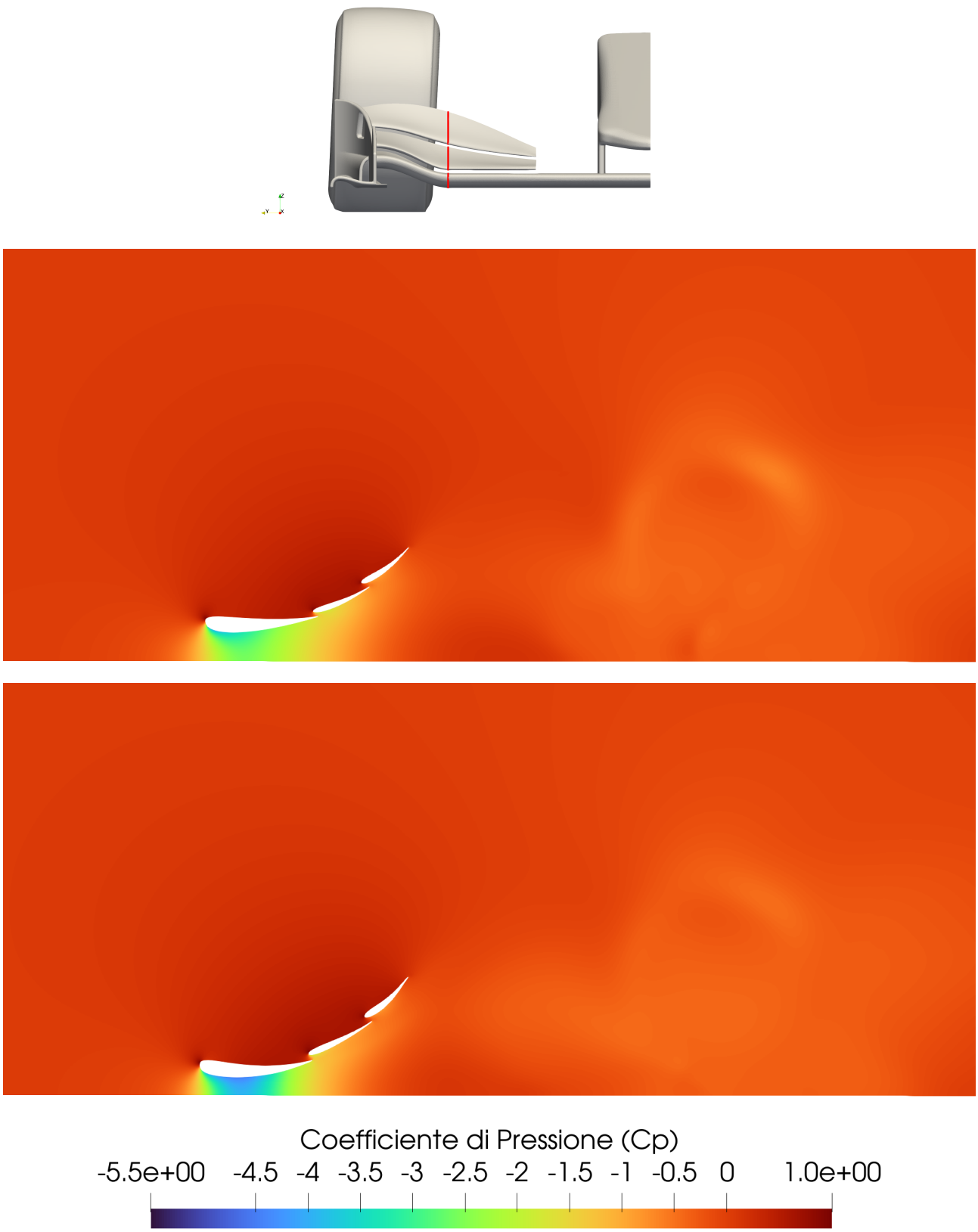


Figura 7.33: Contour di C_p - Geometria originale vs Geometria modificata - $y/s = 0.63$

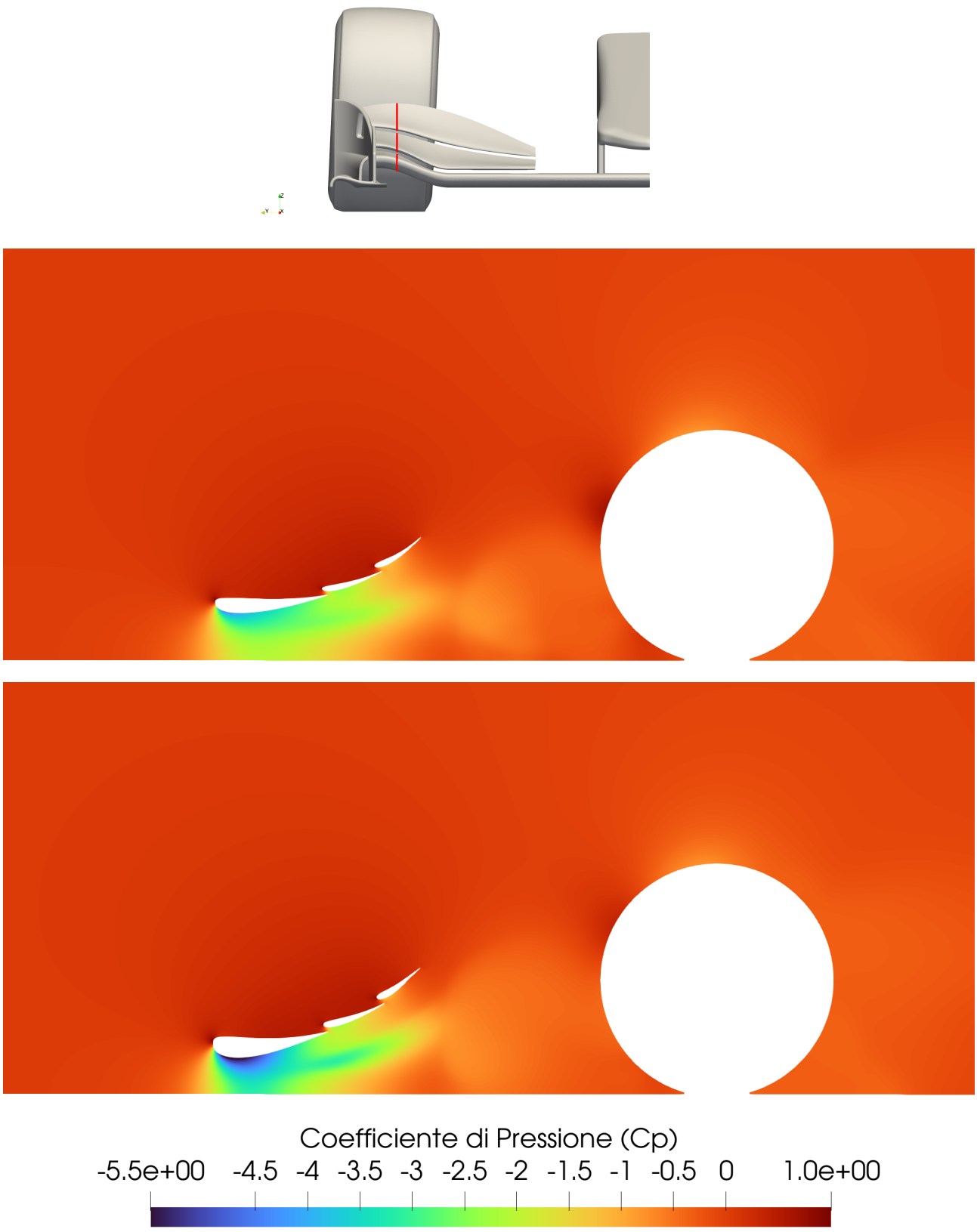


Figura 7.34: Contour di C_p - Geometria originale vs Geometria modificata - $y/s = 0.79$

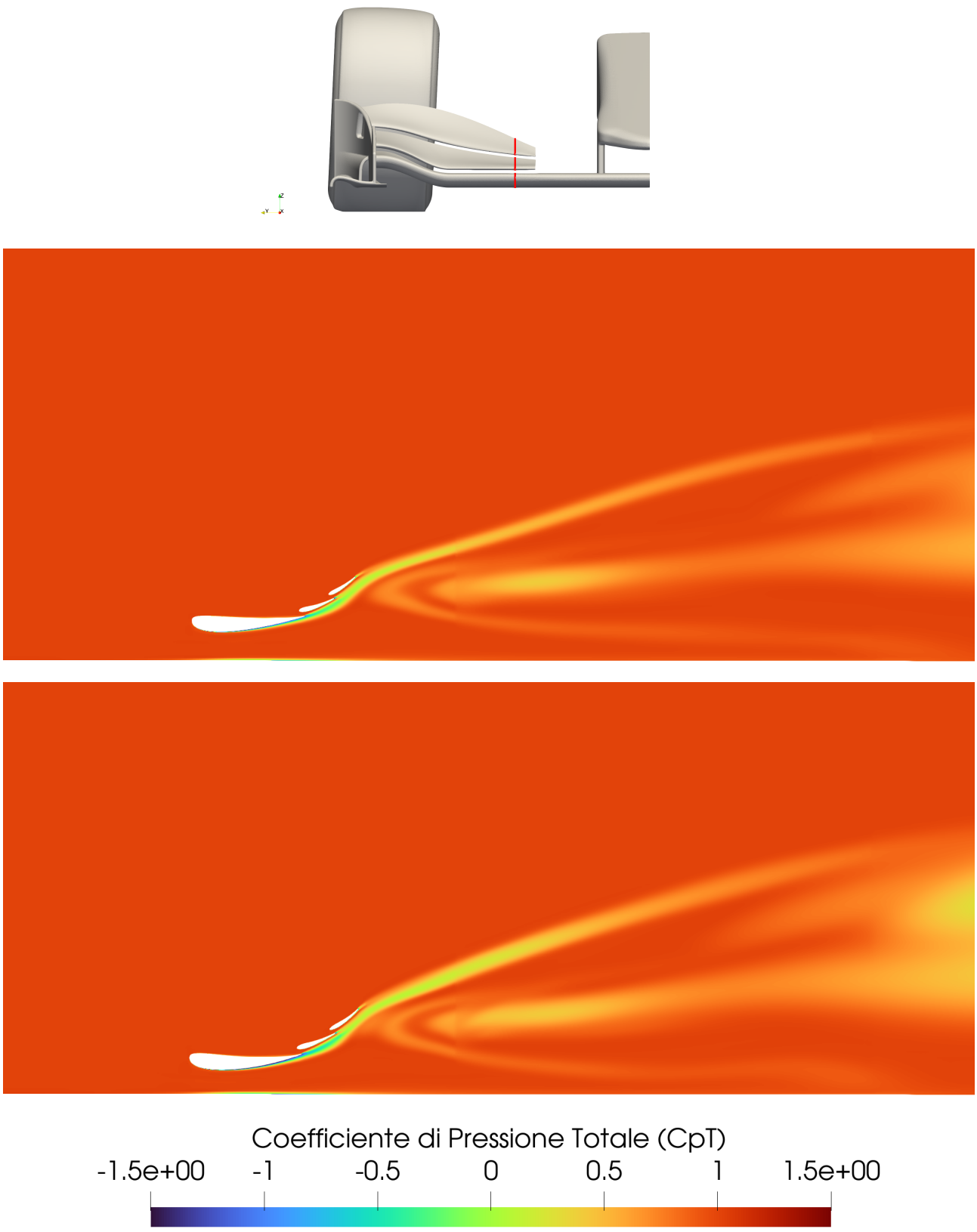


Figura 7.35: Contour di C_{pT} - Geometria originale vs Geometria modificata - $y/s = 0.42$

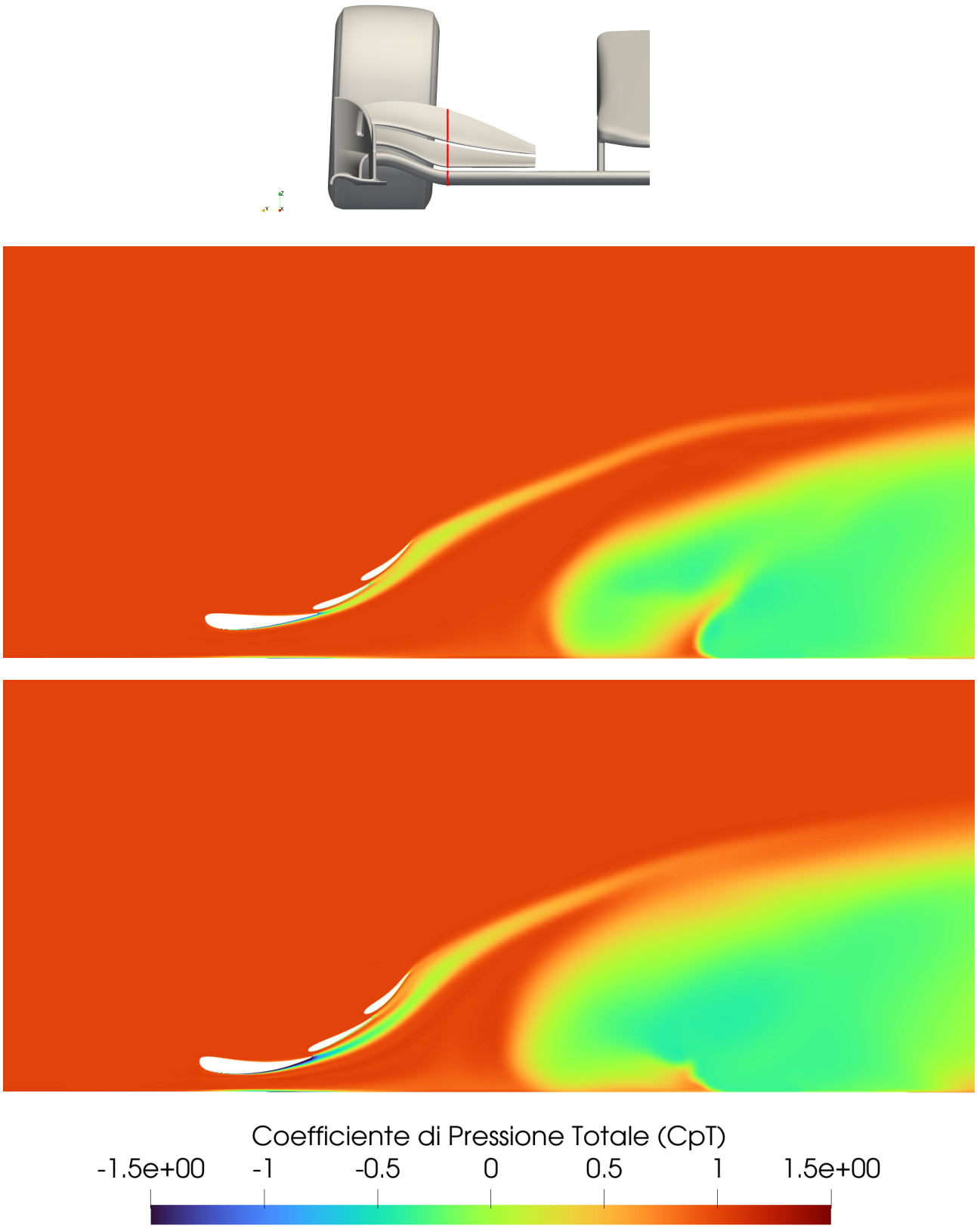


Figura 7.36: Contour di C_{pT} - Geometria originale vs Geometria modificata - $y/s = 0.63$

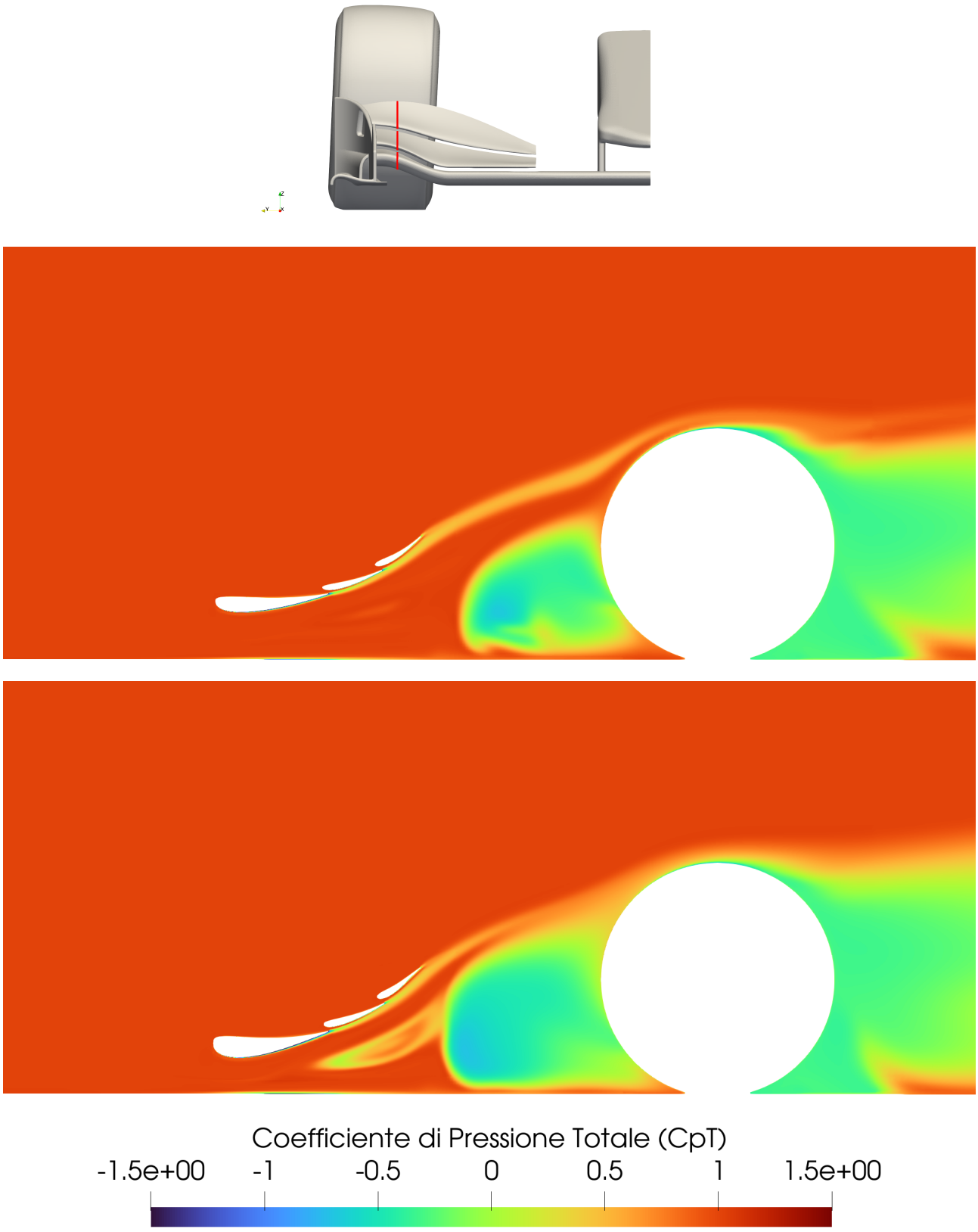


Figura 7.37: Contour di C_{pT} - Geometria originale vs Geometria modificata - $y/s = 0.79$

Capitolo 8

Conclusioni

Il lavoro di tesi si è posto come obiettivo quello di analizzare il sistema costituito da ala e ruota anteriore di una vettura tipo formula utilizzando OpenFOAM, un software open source per la fluidodinamica computazionale. Una volta portato a termine questo primo passaggio il secondo obiettivo è stato ricavare le mappe di sensitività della geometria, rispetto a delle funzioni obiettivo prestabilite, ed ottimizzare la geometria dell'ala, per massimizzarne la deportanza, variandone la forma. Il lavoro è servito anche per confrontare OpenFOAM con un altro software ampiamente utilizzato in ambito industriale, non open source e quindi sotto licenza, Ansys Fluent.

Sono state incontrate una serie di prime difficoltà nell'ottenere risultati accettabili legate alla forte instazionarietà del caso, data sicuramente dalla presenza della ruota che, in quanto corpo tozzo, genera dietro di se una forte scia, con strutture vorticose che variano fortemente nel tempo. Dopo aver modificato le griglie di calcolo e il setup si è riusciti ad ottenere degli ottimi risultati. Questi, sono stati confrontati con quelli ottenuti con Ansys Fluent, rivelandosi quasi perfettamente uguali in termini di distribuzioni del coefficiente di pressione sui flap dell'ala.

Si è passati poi al calcolo dell'aggiunto, qui si sono incontrate molte difficoltà iniziali, sempre legate all'instazionarietà del caso, nel portare il calcolo a convergenza. Grazie all'aiuto del Politecnico di Atene, responsabile dell'implementazione dell'aggiunto continuo in OpenFOAM, sono state apportate modifiche al setup che hanno aiutato ad ottenere una buona convergenza dei calcoli. Per ottenere dei risultati più stabili, condizione imprescindibile per ottenere una buona convergenza dell'aggiunto, si è deciso di provare ad utilizzare il modello di turbolenza Spalart-Allmaras al posto di $k-\omega$ SST, nonostante non fosse il modello di turbolenza più adatto al caso in esame. Questo, è stato fatto in quanto solo con Spalart-Allmaras si è riusciti ad ottenere una convergenza dell'aggiunto con il modello di turbolenza "adjointSpalartAllmaras" attivo, mentre, a partire dai risultati diretti ottenuti con $k-\omega$ SST, l'aggiunto è andato a convergenza solo utilizzando "adjointLaminar" che non calcola le grandezze caratteristiche aggiunte della turbolenza, "ka" e "wa".

Una volta verificato che Spalart-Allmaras fornisse una buona convergenza di calcolo diretto e aggiunto si sono confrontati i risultati ottenuti dal calcolo iniziale con quelli ricavati utilizzando il modello di turbolenza $k-\omega$ SST, per verificare che non fossero troppo diversi tra loro. Si sono osservate delle differenze nelle distribuzioni del "C_p" sui vari flap, ma non particolarmente incisive, quindi, al fine dell'ottimizzazione della forma dell'ala, ritenute accettabili.

Una volta ottenute e analizzate le mappe di sensitività è stato creato un setup per effettuare l'ottimizzazione di forma finale, utilizzando tutti e due i modelli di turbolenza studiati. Osservando l'andamento del valore della funzione obiettivo scelta, $J = C_l$, al crescere degli step di ottimizzazione si sono osservati due risultati distinti:

- Con il modello di turbolenza $k - \omega$ SST e utilizzando "adjointLaminar" per la risoluzione dell'aggiunto il valore di "J" diverge dopo i primi passi di deformazione della geometria, quindi i risultati non sono stati considerati accettabili.
- Con il modello di turbolenza Spalart-Allmaras e utilizzando per il calcolo dell'aggiunto "adjointSpalartAllmaras" il valore di "J" non diverge nelle prime iterazioni, bensì continua a decrescere come desiderato. Per le analisi e i confronti finali dei risultati sono quindi stati presi in considerazione solo quelli ottenuti con questo metodo.

Sono stati eseguiti 9 passi di ottimizzazione con come obiettivo la minimizzazione del Coefficiente di Lift (C_l) dell'ala, che ricordiamo essere già negativo in quanto il "lift" ha uguale direzione ma verso opposto a quello dell'asse "z", minimizzandolo, quindi, si sta cercando di massimizzare la deportanza generata dall'ala. Si è passati da un "C_l" pari a -0.984 a -1.137, la "downforce" generata dall'ala è quindi aumentata del 15,57%, confermando la bontà del metodo utilizzato e lasciando aperte molte porte per ottimizzazioni successive con diverse funzioni obiettivo.

Il lavoro di tesi è quindi riuscito ad acquisire e fornire a Dallara Automobili un metodo validato e funzionante per l'ottimizzazione di forma di una geometria complessa, come quella esaminata, utilizzando un software open source quindi non sottoposto a licenza. Successivamente in figura 8.1 e figura 8.2 è possibile osservare geometria originale e geometria ottimizzata per massimizzare la deportanza generata dall'ala.



Figura 8.1: Vista frontale geometria originale



Figura 8.2: Vista frontale geometria modificata

Sviluppi futuri

In merito agli sviluppi futuri a partire da questo lavoro di tesi sarà sicuramente necessario consolidare il setup di calcolo in modo da ottenere risultati corretti e stabili, indipendentemente dal modello di turbolenza utilizzato e dalle condizioni del flusso. Potrebbe inoltre essere interessante variare i parametri di controllo della deformazione, aumentando il numero di punti o il grado del polinomio, secondo cui la griglia di calcolo viene deformata, potrebbe essere possibile effettuare deformazioni più dettagliate e ottenere comunque dei buoni risultati senza stravolgere la geometria di partenza. Bisogna infatti considerare che i campionati automobilistici, solitamente, hanno regole molto stringenti sulle dimensioni e sugli ingombri delle ali delle vetture. Quindi è possibile che debbano essere considerati anche più vincoli rispetto a quelli imposti all'interno di questo lavoro di tesi. Inoltre, sarebbe sicuramente interessante, una volta validato un setup funzionante in tutte le casistiche, effettuare una ottimizzazione della vettura completa di tutte le sue componenti, oppure, verificare i risultati ottenuti in galleria del vento mediante la creazione di un modello fisico della nuova ala, per verificare la bontà dei risultati.

Bibliografia

[1] J. Katz. *Race Car Aerodynamics, Designed for Speed*. BentleyPublishers, 1995.

[2] Giampaolo dallara: un talento tutto made in motor valley. https://www.motorvalley.it/storie/giampaolo-dallara-talento-made-in-motor-valley/.

[3] Dallara story: dal garage allo spazio, un'avventura lunga 50 anni. https://www.quattroruote.it/news/eventi/2022/03/17/dallara_50_anni_storia_dell_azienda.html, 17 Marzo 2022.

[4] Storia di dallara. https://www.dallara.it/it/azienda/storia.

[5] M. Darwish F. Moukalled, L. Mangani. *The Finite Volume Method in Computational Fluid Dynamics*. Springer, 2015.

[6] About openfoam. https://cfd.direct/openfoam/about.

[7] Joaquim R. R. A. Martins Kevin J. Maki Ping He, Charles A. Mader. Dafoam: An open-source adjoint framework for multidisciplinary design optimization with openfoam. *AIAA Journal*, 58, 10 Gennaio 2020.

[8] Stephen B. Pope. *Turbulent Flows*. MPG Books Group, 2000.

[9] Ansys help theory, chapter 4: Turbulence. https://ansyshelp.ansys.com/public/account/secured?returnurl=Views/Secured/corp/v242/en/cfx_mod/i1345898.html.

[10] David C. Wilcox. *Turbulence Modeling for CFD*. 1993.

[11] R. Langtry F. R. Menter, M. Kuntz. Ten years of industrial experience with the sst turbulence model. *Heat and Mass Transfer*, 4:625–632, Gennaio 2003.

[12] S. R. Allmaras P. R. Spalart. A one-equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 1:5–21, 1994.

[13] Y+ wall distance estimation. https://www.cfd-online.com/Tools/yplus.php.

[14] Calculator for the estimation of turbulence properties values (boundary and initial conditions). https://www.wolfdynamics.com/tools.html?id=110.

[15] Turbulence free-stream boundary conditions. https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions.

[16] The spalart-allmaras turbulence model. https://turbmodels.larc.nasa.gov/spalart.html.

[17] Jan Pralits. Slides del corso di ottimizzazione computazionale in fluidodinamica. 2020.

[18] U. Nilsson. *Description of adjointShapeOptimizationFoam and how to implement new objective functions*. 20 Gennaio 2014.

[19] Eric Segalerba. *Ottimizzazione Aerodinamica dell'ala frontale per una monoposto di Formula 1 del 2022, utilizzando il metodo dell'aggiunto*. Marzo 2021.

[20] N. Galanos K.C. Giannakoglou, E.M. Papoutsis-Kiachagias. *USER MANUAL "adjointOptimisationFoam, an OpenFOAM-based optimisation tool"*. Dicembre 2023.

[21] Community contribution: New adjoint to k-omega sst model. `https://www.cfd-online.com/Tools/yplus.php`, 2022.