

**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

**FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE**

**CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA AEROSPAZIALE**

Curriculum Aeronautico

**Ottimizzazione Aerodinamica di un
velivolo mediante CAD Parametrico
e Mesh Morphing**

Relatori:

Prof. FEDERICO DALLA BARBA

Prof. MARCO E. BIANCOLINI

Correlatore:

Ing. ANDREA LOPEZ

Laureando:

GIANLUCA MAGRI'

ANNO ACCADEMICO 2022-2023

Sommario

Il campo della simulazione svolge un ruolo sempre più centrale nella fase di progettazione industriale, consentendo di ridurre i tempi di sviluppo e i costi per i prototipi. In particolare, in questo elaborato si è studiata l'interazione tra strumenti di analisi fluidodinamica computazionale, mesh morphing e CAD (Computer-Aided Design) parametrico al fine di effettuare un'ottimizzazione aerodinamica di un velivolo. La collaborazione con l'azienda RBF-MorphTM ha reso possibile l'analisi parametrica di un condotto di test e del successivo velivolo in modo da valutare al meglio la forma ottimale per migliorare le prestazioni. La riduzione della caduta di pressione interna e l'aumento dell'efficienza aerodinamica sono stati i due principali obiettivi, rispettivamente, per l'ottimizzazione del condotto e dell'aereo di linea. I modelli CAD sono stati creati utilizzando CadQuery, una libreria di Python, ed il software Engineering Sketch Pad (ESP). Si è fatto uso di Ansys Fluent come software CFD e di RBF Morph per eseguire le operazioni di mesh morphing. Infine, sono stati creati codici in linguaggio Python per automatizzare le procedure. Una volta ottenuta la geometria più efficiente tramite la metodologia delle superfici di risposta, tale configurazione è stata verificata con un'ulteriore analisi CFD. I risultati ottenuti sono stati conformi e ragionevoli dal punto di vista fisico. La velocità e la possibilità di automatizzazione del processo sono i punti fondamentali delle tecnologie software utilizzate. Considerando, infine, licenze software complete e capacità di calcolo superiori, l'ottimizzazione di forma può essere effettuata in tempi davvero rapidi.

Indice

Abstract	2
1 Introduzione	11
1.1 L'analisi CFD	12
1.1.1 Volume di controllo e sistema	13
1.1.2 Conservazione della massa	14
1.1.3 Bilancio della quantità di moto	15
1.1.4 Bilancio dell'energia	15
1.1.5 Legge dei gas ideali	16
1.1.6 Turbolenza	17
1.1.7 Classificazione della mesh	22
1.1.8 Qualità della mesh	24
1.2 Ottimizzazione	25
1.2.1 Design of Experiment	25
1.2.2 Formulazione generale di un problema di ottimizzazione	27
1.2.3 Classificazione dei metodi di ottimizzazione	29
1.3 Parametrizzazione di forma e mesh morphing	30
1.3.1 Il CAD parametrico	31
1.3.2 Mesh Morphing	31
1.3.3 Radial Basis Functions	32

2	Strumenti	35
2.1	CadQuery	35
2.2	Ansys Workbench	37
2.3	Ansys Fluent	39
2.4	RBF Morph Fluids	43
2.5	Engineering Sketch Pad	46
3	Caso di studio: condotto di test	48
3.1	Generazione del CAD di base	48
3.2	Creazione del DOE	50
3.3	Creazione della mesh	53
3.4	Fase di set-up e soluzione	57
3.5	Risultati baseline	59
3.6	Workflow completo - condotto	61
3.7	Condotto di esempio	68
3.8	Ottimizzazione	70
4	Caso di studio: aeromobile civile	76
4.1	Configurazione base del velivolo	76
4.2	Creazione del DOE	79
4.3	Preparazione della geometria e della mesh	81
4.4	Simulazione baseline	84
4.5	Workflow completo - velivolo	93
4.6	Modello di esempio	96
4.7	Ottimizzazione	100
5	Conclusioni	108
	Bibliografia	111

Appendice A	112
Appendice B	117
Ringraziamenti	123

Elenco delle figure

1.1	Decomposizione di Reynolds della grandezza "v"	18
1.2	Tipi di Mesh strutturate	23
1.3	Mesh non strutturata di un profilo alare 2D	23
1.4	Mesh ibrida di un profilo alare 2D	24
1.5	Latin hypercube sampling	26
1.6	Esempio di una superficie di risposta 3D	30
1.7	Punti sorgente (in rosso) e punti traslati (in verde)	33
2.1	Lastra solida ottenuta con CadQuery	37
2.2	Interfaccia Ansys Workbench	38
2.3	Fasi della simulazione CFD	39
2.4	Interfaccia di Fluent	40
2.5	Post-processing	42
2.6	Interfaccia di RBF Morph	43
2.7	Esempi di sistemi aeronautici ottenuti con ESP	47
3.1	CAD baseline del condotto	50
3.2	File csv con tutti i design points	52
3.3	Files .step	53
3.4	Studio di convergenza	54

3.5	Mesh utilizzata per lo studio di ottimizzazione	56
3.6	grafico di v_out	59
3.7	Linee di contorno della velocità all'uscita	59
3.8	Grafico dei residui	60
3.9	cfg.txt	63
3.10	journal.scm	64
3.11	mod.pts	65
3.12	Geometrie a confronto: a sinistra la baseline, a destra il DP in questione	68
3.13	punti sorgente e punti finali	69
3.14	Confronto tra mesh modificata (in azzurro) e mesh originale	70
3.15	Strumento di Ottimizzazione	71
3.17	Superfici di risposta del condotto	73
3.18	Impostazione di ottimizzazione	74
3.19	Punti candidati	74
3.20	Sensibilità locale	75
4.1	Modello di Boeing 787-800	77
4.2	Modello modificato	77
4.3	Valori dei parametri per i primi 19 DP	80
4.4	Enclosure	81
4.5	Dominio di calcolo	82
4.6	Mesh utilizzata	84
4.7	Residui	89
4.8	Parametri di controllo: Cd e portata all'Outlet	89
4.9	Linee di contorno - zone principali	90
4.10	cfg.txt - velivolo	94
4.11	journal.scm - velivolo	95

4.12 Geometrie a confronto: in alto la baseline, in basso il DP1	96
4.13 punti sorgente e punti finali	97
4.14 griglie a confronto	98
4.15 results.csv	100
4.16 risultati importati nel DOE del blocco "Response Surface Optimization" del Workbench	100
4.17 distribuzione delle ampiezze	101
4.18 Obiettivo: massimizzare l'efficienza	101
4.19 punti candidati - velivolo	101
4.20 Alcune superfici di risposta 3D	102
4.21 Sensibilità locale	103
4.22 a sinistra la geometria baseline, a destra la geometria ottimizzata . . .	104
4.23 campo di pressione e velocità per un profilo alare prossimo alla radice dell'ala	105
4.24 campo di pressione e velocità per la fusoliera: baseline a sinistra, ottimo a destra	106

Elenco delle tabelle

1.1	coefficienti del modello k-Epsilon	21
3.1	Caratteristiche delle griglie per lo studio di convergenza	55
3.2	Fattori di qualità	55
3.3	Metodi di soluzione	58
4.1	Caratteristiche del modello e del velivolo reale	78
4.2	Range di valori utilizzati	79
4.3	Caratteristiche della mesh	83
4.4	Dati dell'aria a 7000 ft	85
4.5	Metodi di soluzione	86
4.6	Strategia di convergenza	88

Capitolo 1

Introduzione

Tra le applicazioni software che agevolano la risoluzione di problemi tecnologici tramite il calcolo numerico, indicate con la sigla CAE (Computer-aided engineering), la modellazione CFD (Computational Fluid Dynamics) è uno strumento fondamentale per molte realtà industriali. Per molti anni la progettazione aerodinamica prevedeva esclusivamente lo svolgimento di test in galleria del vento. Ad oggi, invece, le simulazioni CFD consentono di risparmiare tempo nelle fasi di realizzazione di prototipi in scala, prevedendo le prestazioni del prodotto finale, le quali vengono confermate dai dati in galleria del vento. La ricerca della geometria aerodinamica ottimale è ancora soggetta a tempi di sviluppo significativi. L'utilizzo di tecniche di "mesh morphing" permette di analizzare in maniera più rapida gli effetti causati dalla variazione geometrica. L'integrazione di questi strumenti con un modello CAD (computer-aided design) semplice o parametrico e un ambiente di ottimizzazione numerica garantisce una rapida analisi. Lo scopo di questo elaborato è svolgere un'ottimizzazione di forma di un condotto semplice e, successivamente, estendere il procedimento ad un modello di aereo civile.

1.1 L'analisi CFD

Tutte le problematiche che coinvolgono l'azione di fluidi vengono affrontate tramite analisi CFD. I campi di applicazione sono numerosi e includono molti settori industriali. Le equazioni di Navier-Stokes che governano il moto dei fluidi (gas e liquidi) sono un sistema di equazioni differenziali alle derivate parziali non lineari. Tale sistema di equazioni deriva dalla conservazione della massa, della quantità di moto e dell'energia. La soluzione analitica a questo sistema si ottiene solo per casi semplici. Per tutti gli altri casi l'unico modo è risolvere il sistema numericamente. Un passo fondamentale è definire il dominio del fluido tramite uno spazio discretizzato. Il volume viene diviso in celle (elementi) generando così una griglia di calcolo, denominata "mesh". La mesh determina il passo di integrazione spaziale per la soluzione delle equazioni discretizzate che descrivono il comportamento del sistema. Celle di dimensioni inferiori permettono di risolvere in modo più preciso le equazioni di Navier-Stokes, richiedendo però un costo computazionale maggiore. In generale, per risolvere un problema fluidodinamico si segue una procedura standardizzata. Tale struttura viene utilizzata dalla maggior parte dei software commerciali e prevede:

- Discretizzazione del dominio, creazione della mesh
- Scelta del modello di fluido, tipo di simulazione e modello di turbolenza
- Definizione delle condizioni al contorno
- Scelta dei parametri del solutore
- Simulazione numerica
- Post-processing, valutazione e verifica dei risultati ottenuti

1.1.1 Volume di controllo e sistema

La descrizione del comportamento di un fluido deriva da alcune leggi fisiche fondamentali le quali sono espresse attraverso un sistema di equazioni. In particolare, l'analisi di un campo fluidodinamico si fonda sulla conservazione della massa, sulla seconda legge di Newton e sul primo principio della termodinamica. Queste equazioni fondamentali si applicano in modi diversi facendo uso dei concetti di sistema e di volume di controllo. Il sistema è un insieme di materia contenente un numero specifico di particelle di fluido e tale sistema si può muovere ed interagire con l'ambiente circostante. Il volume di controllo, invece, è un volume dello spazio dove scorre del fluido. In molti casi si è interessati a valutare le azioni esercitate da un fluido su un oggetto; si utilizza il concetto di volume di controllo che includa l'oggetto. Per l'analisi del moto del fluido è possibile seguire due metodi diversi: il primo, detto *euleriano*, utilizza il concetto di campo e analizza il moto del fluido descrivendone tutte le proprietà caratteristiche (pressione, densità, velocità, ecc.) in funzione delle coordinate temporali e spaziali. Le caratteristiche sul moto dipendono da ciò che avviene in certi punti fissi dello spazio mentre il fluido scorre. Il secondo tipo di approccio, detto *lagrangiano*, utilizza delle particelle di fluido individuali seguendole nel loro moto e valutando come cambiano nel tempo le proprietà del fluido ad esse associate. Di solito, in fluidodinamica è più utile adottare una descrizione euleriana del fenomeno. In poche parole, nel sistema (descrizione lagrangiana) si segue il fluido e lo si studia nel tempo. Nel volume di controllo (descrizione euleriana), invece, si osserva il fluido in una porzione dello spazio. Tutte le leggi fondamentali del moto sono espresse nella forma originale considerando un sistema fisico.

Per utilizzare le equazioni dal punto di vista applicativo è necessario riformularle in termini di volume di controllo. Si introduce a tal fine il teorema del trasporto di

Reynolds:

$$\frac{d}{dt} B_s(t) = \frac{d}{dt} \int_{V_c} \rho b dV + \int_{\delta V_c} \rho b \vec{V} \cdot \hat{n} dS \quad (1.1)$$

dove con B si indica una qualsiasi grandezza estensiva, ovvero proporzionale alla quantità di materia. b è la grandezza intensiva, cioè $b = B/m$. L'equazione collega la variazione nel tempo della grandezza del sistema con la variazione nel tempo della grandezza nel volume di controllo tramite un termine di flusso calcolato sulla superficie di controllo δV_c .

Per ricavare la forma differenziale delle equazioni del moto si ricorda il teorema di Green (o teorema della divergenza):

$$\int_{\delta V} \vec{C} \cdot \hat{n} dS = \int_V \vec{\nabla} \cdot \vec{C} dV \quad (1.2)$$

La relazione permette di trasformare l'integrale di superficie del flusso di un campo vettoriale \vec{C} in un integrale di volume della divergenza di quella grandezza vettoriale.

1.1.2 Conservazione della massa

La prima legge alla base delle equazioni di Navier-Stokes è sicuramente la conservazione della massa, detta anche equazione di continuità. Essa prevede che, poiché la massa di un sistema si conserva, la variazione nel tempo della massa sia nulla. Di seguito viene riportata la forma integrale (1.3) e la forma differenziale (1.4).

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \vec{V} \cdot \hat{n} dS = 0 \quad (1.3)$$

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0 \quad (1.4)$$

\vec{V} è il vettore della velocità, ρ è la densità del fluido. In caso di stazionarietà, la variazione nel tempo della massa contenuta nel volume di controllo è nulla e, quindi,

la portata in massa uscente eguaglia quella entrante.

1.1.3 Bilancio della quantità di moto

La seconda legge della meccanica si esprime:

$$\sum \vec{F}_{ext} = m \vec{a} = m \frac{d\vec{V}}{dt} \quad (1.5)$$

Se consideriamo un sistema tridimensionale questa equazione produce un'equazione per ogni direzione (x,y,z). Applicando il teorema del trasporto e della divergenza all'equazione vettoriale (1.5), si ottiene:

$$\frac{\partial(\rho\vec{V})}{\partial t} + \vec{\nabla} \cdot (\rho\vec{V} \times \vec{V}) = -\vec{\nabla}p + \vec{\nabla} \cdot \sigma + \rho\vec{f} \quad (1.6)$$

In questa equazione compaiono le forze di pressione ∇p , le forze esterne per unità di massa \vec{f} e le forze viscosi. σ è il tensore degli sforzi viscosi e viene definito come:

$$\sigma = \mu(\vec{\nabla}\vec{V} + \nabla\vec{V}^T) + \lambda\vec{\nabla} \cdot \vec{V}I \quad (1.7)$$

μ è la viscosità dinamica, λ è la viscosità volumetrica e I è il tensore identità.

1.1.4 Bilancio dell'energia

Per completare il sistema di equazioni della fluidodinamica è necessario ricavare l'equazione di conservazione dell'energia. A tal fine si ricorre al primo principio della termodinamica:

$$\Delta E_T = Q + L \quad (1.8)$$

dove ΔE_T indica la variazione dell'energia totale del sistema, la quale è provocata dall'aumento di energia per effetto del calore Q e dall'aumento di energia fornito dal

lavoro esterno L sul sistema. La stessa relazione può essere riscritta riferendosi a variazioni dell'energia nell'unità di tempo. Ogni termine avrà la dimensione di una potenza. Dopo una serie di calcoli e l'applicazione dei teoremi già discussi si giunge alla seguente equazione:

$$\frac{\partial(\rho E)}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V} E) = -\vec{\nabla} \cdot (p \vec{V}) + \vec{\nabla} \cdot (\sigma \cdot \vec{V}) + \rho \vec{f} \cdot \vec{V} + \vec{\nabla} \cdot (k \nabla T) + Q \quad (1.9)$$

La variabile E è l'energia totale per unità di massa ed è data dalla somma dell'energia cinetica e dell'energia interna $E = 1/2 V^2 + e$. Il termine Q identifica il calore interno mentre il penultimo termine la conduzione di calore, in cui k indica la conduttività termica. I termini restanti simboleggiano il lavoro delle varie forze già comparse nel bilancio della quantità di moto.

1.1.5 Legge dei gas ideali

I tre principi fisici descritti nelle sezioni precedenti generano un sistema di 5 equazioni: la conservazione della massa e dell'energia sono equazioni scalari mentre la quantità di moto è un'equazione vettoriale. Le incognite del sistema sono 7: densità, pressione, temperatura, le tre componenti della velocità e l'energia interna e . Con 5 equazioni e 7 incognite il problema non è risolvibile. La chiusura del sistema è possibile utilizzando le equazioni di stato per pressione, temperatura, densità ed energia interna. Per un gas ideale si ha:

$$p = \rho R T \quad (1.10)$$

$$e = c_v T \quad (1.11)$$

con R costante specifica dei gas e c_v il calore specifico a volume costante.

Per approfondimenti consultare [3].

1.1.6 Turbolenza

Anche con condizioni al contorno stazionarie, le grandezze vettoriali e scalari che caratterizzano il sistema non sono stazionarie ma mostrano oscillazioni casuali attorno ad un valore medio che, in generale, può variare nel tempo. Il moto turbolento è completamente descritto dalle equazioni di Navier-Stokes le quali sono in grado di definire il moto dalle scale macroscopiche a quelle microscopiche. Esistono fondamentalmente tre tipi di approcci: DNS, LES e RANS.

L'approccio DNS (Direct Numerical Simulation) prevede la risoluzione diretta delle equazioni nella loro forma originaria. Tale tecnica richiede una discretizzazione spaziale e temporale molto accurata tale da risolvere anche le più piccole scale dei fenomeni in gioco. La DNS viene utilizzata per flussi a basso numero di Reynolds¹ perché all'aumentare di esso si riducono le dimensioni delle strutture vorticosi più piccole ed il costo computazionale delle simulazioni diventa dunque eccessivo.

Un altro metodo è rappresentato dalla LES (Large Eddy Simulation), mediante cui vengono simulate in maniera esatta le grandi strutture vorticosi, mentre viene modellato il contributo delle piccole scale. Si definisce un limite di separazione, denominato *filtro*, che divide le scale modellate da quelle risolte esattamente.

La conoscenza dettagliata delle quantità istantanee non è sempre necessaria per molte applicazioni ingegneristiche. Molto spesso risulta sufficiente conoscere i valori medi delle grandezze di interesse. Le equazioni RANS (Reynolds Averaged Navier-Stokes) hanno proprio questo scopo: mediare tutte le equazioni del moto in modo da trascurare ed eliminare tutte le oscillazioni turbolente istantanee e poco rilevanti dal punto di vista temporale rispetto al lasso di tempo che si vuole osservare. Ulteriori dettagli nei testi di riferimento [5], [4].

¹Definito come: $Re = \frac{vL}{\nu}$; è un numero adimensionale che misura il rapporto tra le forze d'inerzia e le forze viscosi di un flusso

Mediante la decomposizione di Reynolds:

$$u(x, t) = U(x, t) + u'(x, t) \quad (1.12)$$

essendo u il campo di velocità istantaneo, esso si può scomporre in un termine medio U e un termine di fluttuazione u' [fig.1.1]. Mediare l'equazione (1.12) dimostra banalmente che la media della fluttuazione è sempre nulla. Applicando questa de-

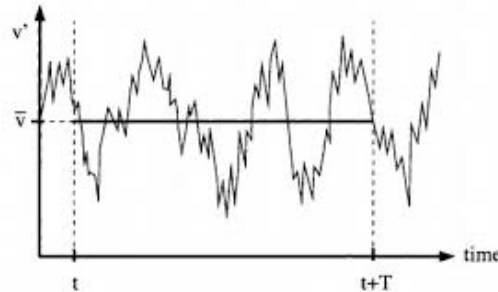


Figura 1.1: Decomposizione di Reynolds della grandezza "v"

composizione alle equazioni di Navier-Stokes incomprimibili, ovvero flusso a densità costante, e assumendo fluidi di tipo Newtoniano², si ottiene il seguente sistema:

$$\begin{cases} \vec{\nabla} \cdot \vec{U} = 0 \\ \frac{D\vec{U}}{Dt} = -\frac{\vec{\nabla} p}{\rho} + f + \mu \nabla^2 \vec{V} - \vec{\nabla} \cdot \langle \vec{u}' \vec{u}' \rangle \end{cases} \quad (1.13)$$

Il termine $\langle \vec{u}' \vec{u}' \rangle$ rappresenta i cosiddetti sforzi di Reynolds mediante i quali è possibile studiare il fenomeno della turbolenza. E' stato dimostrato che la componente $-\vec{\nabla} \cdot \langle \vec{u}' \vec{u}' \rangle$ è statisticamente negativa e compare nell'equazione dell'energia del campo medio sottraendo energia, la quale è trasferita come energia di fluttuazione.

² fluido con viscosità costante in cui le tensioni dipendono linearmente dalla velocità di deformazione

Si può quindi scrivere:

$$k = \frac{\langle u'_i u'_i \rangle}{2} \quad (1.14)$$

$$K = \frac{U_i U_i}{2} \quad (1.15)$$

dove k è l'energia cinetica turbolenta o di fluttuazione e K è l'energia cinetica di campo medio. Riscrivendo l'equazione della quantità di moto si può esprimere la derivata temporale di K (1.16):

$$\frac{DK}{Dt} = \vec{\nabla} \cdot \left[-\frac{p}{\rho} \vec{U} + 2\nu E \cdot \vec{U} - \langle \vec{u}' \vec{u}' \rangle \cdot \vec{U} \right] - 2\nu E : E + \langle \vec{u}' \vec{u}' \rangle : \vec{\nabla} \vec{U} \quad (1.16)$$

in cui:

$$E = \frac{1}{2} (\vec{\nabla} \vec{U} + \vec{\nabla} \vec{U}^T)$$

In modo sintetico l'equazione (1.16) diventa:

$$\frac{DK}{Dt} = \vec{\nabla} \cdot \vec{F}^K - D - P \quad (1.17)$$

- $\vec{\nabla} \cdot \vec{F}^K$ rappresenta il flusso dell'energia cinetica di campo medio;
- D identifica la dissipazione di energia cinetica da parte delle forze viscosse;
- P è un termine di produzione di energia ma il segno negativo la rende una dissipazione;

Per quanto riguarda l'energia cinetica di fluttuazione invece si arriva a:

$$\frac{\partial k}{\partial t} = C + \vec{\nabla} \cdot \vec{F}^k - \mathcal{E} + P \quad (1.18)$$

- C identifica il trasporto convettivo;

- $\vec{\nabla} \cdot \vec{F}^k$ rappresenta il flusso spaziale di k dovuto a pressione, fenomeni viscosi e fenomeni di fluttuazione turbolenta;
- \mathcal{E} è una dissipazione turbolenta
- P compare con il segno positivo quindi è effettivamente una produzione di energia;

Le equazioni mediate (1.13) sono identiche alle equazioni originali a meno del termine supplementare degli sforzi viscosi. Ricordando che $\vec{u}' = (u', v', w')$, tridimensionalmente questo termine si identifica in un tensore:

$$\begin{bmatrix} \langle u'^2 \rangle & \langle u'v' \rangle & \langle u'w' \rangle \\ \langle u'v' \rangle & \langle v'^2 \rangle & \langle v'w' \rangle \\ \langle u'w' \rangle & \langle v'w' \rangle & \langle w'^2 \rangle \end{bmatrix} \quad (1.19)$$

Essendo una matrice simmetrica, le componenti sono 9 ma le incognite totali sono 6. Se consideriamo tutte le variabili, il sistema è composto da 4 equazioni e 10 incognite. Il problema diventa perciò irrisolvibile in queste condizioni. Si ricorre a dei modelli di turbolenza che introducono nel sistema 6 equazioni aggiuntive per rendere il problema ben posto. Quest'ultime relazioni legano le componenti del tensore alle variabili primarie (velocità, pressione media). I modelli di turbolenza sono vari e non esiste un modello di turbolenza ideale per tutti i casi. Di seguito vengono illustrati i più utilizzati:

- Il modello **Spalart-Allmaras** è ottimo quando si ha a che fare con uno strato limite³ attaccato o leggermente separato. Non è adatto per rappresentare forti circolazioni. Risulta funzionale per tutti i problemi di tipo aerodinamico che includano profili alari.

³zona adiacente alla parete in cui avviene la separazione tra flusso indisturbato e flusso rallentato

- Il modello **k-Epsilon** è un ottimo compromesso tra costo computazionale e accuratezza. Si riescono a riprodurre bene le zone di ricircolo sia in assenza che in presenza di scambio di calore. E' possibile modellare gli sforzi di Reynolds confrontandoli con gli sforzi molecolari. Viene, perciò, introdotta una nuova variabile denominata viscosità turbolenta ν_T . Il metodo consiste nel risolvere un'equazione per l'energia cinetica turbolenta (1.20) e una per la dissipazione turbolenta (1.21):

$$\frac{\partial k}{\partial t} + \vec{\nabla} \cdot (\vec{U}k) = \vec{\nabla} \cdot \left[\frac{\nu_T}{\sigma_k} \vec{\nabla} k \right] - \varepsilon + 2\nu_T EE \quad (1.20)$$

$$\frac{\partial \varepsilon}{\partial t} + \vec{\nabla} \cdot (\varepsilon \vec{U}) = \vec{\nabla} \cdot \left[\frac{\nu_T}{\sigma_\varepsilon} \vec{\nabla} \varepsilon \right] + C_{\varepsilon 1} P \frac{\varepsilon}{k} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} \quad (1.21)$$

Una volta ricavate k e \mathcal{E} si ottiene:

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \quad (1.22)$$

Infine, di seguito si riportano i valori standard utilizzati per i coefficienti introdotti:

Coefficienti di chiusura	
C_μ	0.09
σ_k	1
σ_ε	1.3
$C_{\varepsilon 1}$	1.44
$C_{\varepsilon 2}$	1.92

Tabella 1.1: coefficienti del modello k-Epsilon

- Il modello **k-Omega** è molto simile al modello visto in precedenza. Anch'esso composto da due equazioni di trasporto, si differenzia per la scelta della seconda variabile. Al posto di ε si utilizza $\omega = \varepsilon/k$ che rappresenta la pulsazione

turbolenta. Si ottiene un'equazione simile alla (1.21):

$$\frac{\partial \omega}{\partial t} + \vec{\nabla} \cdot (\omega \vec{U}) = \vec{\nabla} \cdot \left[\frac{\nu_T}{\sigma_\omega} \vec{\nabla} \omega \right] + C_{\omega 1} P \frac{\omega}{k} - C_{\omega 2} \omega^2 \quad (1.23)$$

Questo modello riesce a descrivere in modo più dettagliato gli effetti dovuti ai gradienti di pressione nello strato limite, vicino alla parete.

- Il modello misto **k-Omega SST** utilizza k-Epsilon per il flusso indisturbato e k-Omega per il flusso vicino alla parete. In questo elaborato verrà impiegato proprio questo modello in quanto risulta molto accurato e flessibile.

1.1.7 Classificazione della mesh

Quando si lavora con modelli tridimensionali, le celle volumetriche che costituiscono la mesh possono essere di tipologie diverse. Gli elementi 3D più diffusi sono tetraedri ed esaedri, quest'ultimi solitamente prismi a base equilatera. Le griglie, invece, si dividono in:

1. Mesh strutturate: hanno una disposizione delle celle regolare e ripetitiva. Godono di un preciso controllo del reticolo e, per questo motivo, raggiungono una qualità elevata. Queste mesh richiedono una costruzione interamente manuale e risultano poco adatte per geometrie molto complesse. A loro volta, è possibile un'ulteriore suddivisione:
 - Mesh cartesiane o curvilinee [fig.1.2a]
 - Mesh body-fitted [fig.1.2b], ovvero realizzate sul corpo
 - Mesh Multi-Block [fig.1.2c], in cui il dominio è suddiviso in regioni, ognuna delle quali è una mesh strutturata.

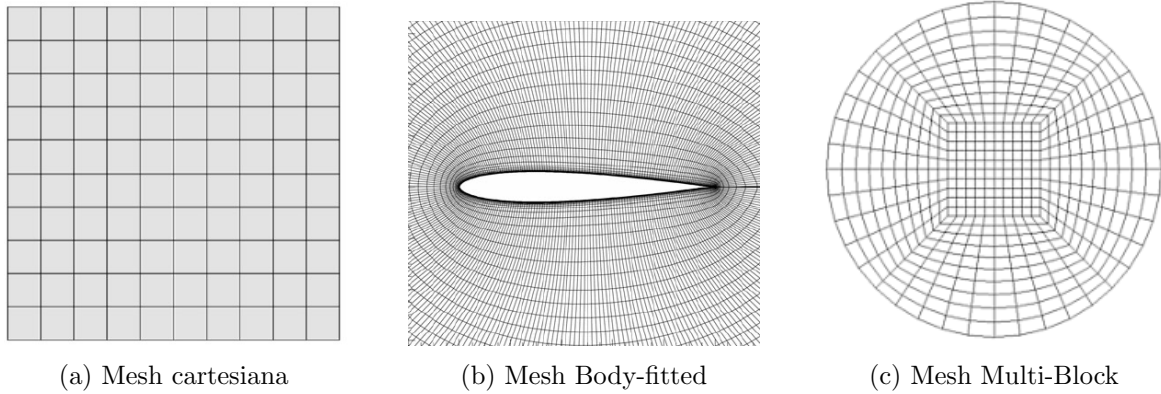


Figura 1.2: Tipi di Mesh strutturate

2. Mesh non strutturate [fig.1.3]: non c'è una struttura ordinata e vengono generate velocemente ed automaticamente da algoritmi di space-filling. Sono adatte a domini complessi e il numero di celle può essere inferiore rispetto alla mesh strutturata equivalente. La qualità può essere minore e il miglioramento può essere difficoltoso.

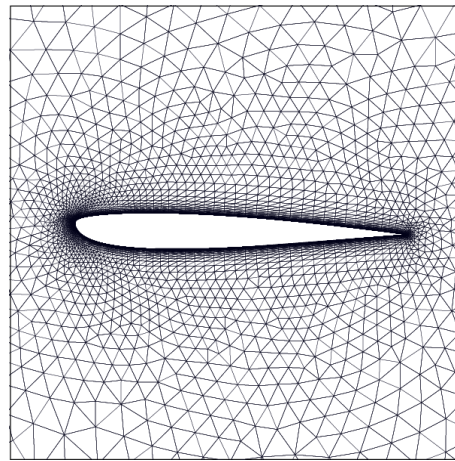


Figura 1.3: Mesh non strutturata di un profilo alare 2D

3. Ibride: combinano le griglie strutturate con quelle non strutturate. Permettono di avere una griglia precisa nelle parti più delicate (vicino alla parete ad esempio) e riducono il numero di celle nelle altre zone. Queste mesh sono un buon compromesso tra accuratezza e tempo di calcolo. L'interfaccia tra i due tipi di

mesh rappresenta un tema critico. Richiedono, infine, un tempo di realizzazione considerevole.

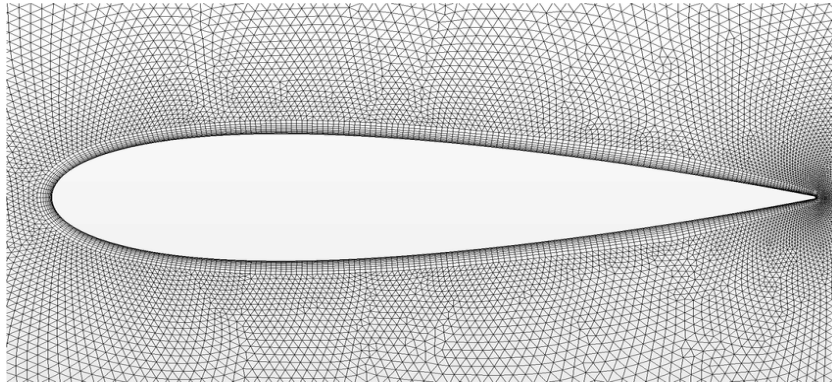


Figura 1.4: Mesh ibrida di un profilo alare 2D

Il tipo di griglia dipende dalla natura del problema da analizzare, dall'accuratezza richiesta, dalla complessità del dominio, dal tipo di solutore e dal tempo a disposizione per costruire la mesh stessa.

1.1.8 Qualità della mesh

La costruzione di una buona mesh è un fattore cruciale per ottenere una buona soluzione. Una griglia di bassa qualità può causare il fallimento del solutore, divergenza o non-convergenza, errori numerici e aumento del tempo di calcolo. Di seguito vengono riportate delle buone pratiche per creare una mesh ottimale.

- Fitta nelle aree vicino a parete e in zone con forti gradienti
- Poco densa nelle aree con bassi gradienti
- Celle poco distorte (equiangle skewness < 0.9)
- Addensamento graduale (Rapporto di area o volume < 10)
- Rapporto d'aspetto contenuto ($\ll 10'000$ circa)

1.2 Ottimizzazione

Una volta definito e progettato l'esperimento, individuate le variabili di processo ed estrapolato i risultati necessari, si procede con l'ottimizzazione dei risultati. Con questo processo si cerca di minimizzare o massimizzare, in base alle richieste, una determinata funzione per ottenere i valori ottimali dei parametri di progetto. Procedendo in maniera ordinata, in primo luogo, verrà analizzata la procedura di progettazione dell'esperimento. Successivamente verranno illustrate le basi per la fase di ottimizzazione.

1.2.1 Design of Experiment

L'interesse nell'investigare il cambiamento di un processo cambiando i dati in ingresso porta a definire una strategia efficiente di sperimentazione. Il Design of Experiment, spesso identificato con l'acronimo DOE, è una metodologia che consente di pianificare un esperimento al fine di dimostrare il legame tra variabili di input e di output di un sistema. Parte integrante di un DOE sono:

- una **risposta**, ovvero un risultato misurabile;
- un **fattore**, una qualsiasi variabile che, si ritiene, possa influenzare la risposta di interesse;
- un **modello** matematico del processo

L'esperimento consiste nell'esercitare il modello su un intervallo di valori assegnati. Un metodo DOE ampiamente utilizzato è quello a progettazione casuale: i valori dei parametri del modello per gli esperimenti vengono assegnati sulla base di un processo random. Tra i metodi random più diffusi rientra sicuramente il cosiddetto **Latin Hypercube Design** (LHD).

Campionamento Latin Hypercube

Il Latin Hypercube Sampling⁴ (LHS) è un metodo statistico per la generazione random di valori dei parametri. Tale metodo può portare a ridurre drasticamente il numero di esecuzioni necessarie per ottenere un risultato ragionevolmente accurato. LHS si basa sul "Latin square design" che rappresenta fundamentalmente un quadrato in cui vi è un singolo campione per ogni riga e colonna, vedi figura 1.5

A	B	C
C	A	B
B	C	A

Figura 1.5: Latin hypercube sampling

Se si considera un problema monodimensionale, Latin Hypercube prevede la divisione della densità di probabilità cumulativa (CDF) in un numero di partizioni uguali in base al numero di elementi richiesti. Viene poi estrapolato un valore casuale da ognuna di queste partizioni, ottenendo così un campione. Nel caso n-dimensionale si procede allo stesso modo: seguendo il caso monodimensionale vengono generati n-campioni che combinati casualmente tra loro forniscono il campione finale.

⁴<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.qmc.LatinHypercube.html>

1.2.2 Formulazione generale di un problema di ottimizzazione

I problemi di ottimizzazione hanno come scopo quello di ricercare punti di lavoro ottimali. Infatti, in ambito ingegneristico si ricercano le condizioni operative per le quali si raggiunge, ad esempio, un massimo rendimento o un buon compromesso tra peso, dimensioni e prestazioni. Un sistema da analizzare può essere non vincolato oppure soggetto a limiti geometrici o fisici. Dal punto di vista matematico, viene definita una funzione di costo in cui vengono tracciati i valori di una o più variabili. I punti di minimo o di massimo (locali o assoluti) vengono chiamati anche punti stazionari. I punti di minimo hanno un piano tangente orizzontale e nel loro intorno vi sono solo punti ad una quota superiore. Viceversa si può dire per i punti di massimo. Esiste anche un'altra categoria di punti stazionari: i cosiddetti punti di sella, identificati sempre da un piano tangente orizzontale, hanno nel loro intorno punti a quota più bassa ed altri punti a quota più alta. Data la funzione generica $f(x)$ continua, il primo passo consiste nella ricerca del piano tangente orizzontale per i punti stazionari. Questa proprietà viene individuata attraverso il gradiente della funzione stessa che in tali punti risulta nullo.

$$\nabla f(x_0) = 0 \quad (1.24)$$

Tuttavia se la funzione non fosse continua o differenziabile in tutto il suo dominio, allora potrebbero esserci punti critici tra cui spigoli e punte i quali non vengono rilevati con la ricerca dei punti stazionari. Una volta individuati i punti, il secondo passo consiste nel distinguerli tra massimi, minimi e selle. Per fare questo, si utilizza la matrice Hessiana, definita qui sotto per una funzione $f(x, y)$:

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y \partial y} \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} \quad (1.25)$$

La matrice H_f è costituita dalle derivate parziali seconde. Per ogni punto stazionario P_0 si sostituiscono i valori e si analizza la matrice. Per capire se si tratta di un punto di massimo, minimo o sella, si utilizza il determinante della matrice Hessiana, detto *Hessiano* e la componente f_{xx} . In particolare:

- P_0 è un massimo se $\det(H_f) > 0$ e $f_{xx} < 0$
- P_0 è un minimo se $\det(H_f) > 0$ e $f_{xx} > 0$
- P_0 è un punto sella se $\det(H_f) < 0$
- se $\det(H_f) = 0$ le informazioni non sono sufficienti

Quando l'ottimizzazione si dice vincolata significa che il dominio è soggetto ad un *vincolo*, ovvero una relazione necessaria tra le variabili oppure un vincolo geometrico. Di conseguenza ci si occupa della ricerca di punti stazionari in un dominio di dimensioni ridotte. Per questo tipo di problema si utilizza il metodo di Lagrange. Identificando con $g(x, y) = 0$ il vincolo di eguaglianza, risulta:

$$L(x, y, \lambda) = f(x, y) + \lambda g(x, y) \tag{1.26}$$

in cui λ è il moltiplicatore di Lagrange ed L la funzione lagrangiana. Applicando il gradiente alla funzione ausiliaria, si ottiene il sistema:

$$\begin{cases} \nabla f(x, y) + \lambda \nabla g(x, y) = 0 \\ g(x, y) = 0 \end{cases} \tag{1.27}$$

il quale se risolto fornisce i punti stazionari che vanno poi interpretati come già visto con la matrice Hessiana H_L .

1.2.3 Classificazione dei metodi di ottimizzazione

Generalmente è possibile distinguere problemi di ottimizzazione globale e locale. I metodi che permettono di ricavare il minimo o il massimo assoluto vengono detti di ordine zero e sono onerosi dal punto di vista computazionale in quanto operano su un dominio molto esteso. D'altra parte, i metodi di ottimizzazione locale, detti anche metodi basati sul gradiente, richiedono meno calcoli e sono più rapidi. Esistono molti metodi differenti sia globali che locali, per ulteriori dettagli consultare [6]. Nel paragrafo seguente viene introdotto il metodo di ordine zero utilizzato nella presente tesi.

Metodologia delle superfici di risposta

La RSM (Response Surface Methodology) è un insieme di tecniche matematiche e statistiche, il cui scopo è esplorare le relazioni tra diverse variabili esplicative di ingresso e una o più variabili di risposta. La superficie di risposta viene creata utilizzando i dati di una ottimizzazione precedente oppure attraverso un DOE, come nel nostro caso. In termini matematici, supponiamo ad esempio che la risposta y sia funzione di due parametri di input:

$$y = f(x_1, x_2) + \epsilon \quad (1.28)$$

dove ϵ rappresenta l'errore numerico. Poiché la relazione che lega y alle due variabili di input è incognita, la tecnica consiste nell'interpolare i punti derivanti dal DOE, determinando così una relazione approssimata tra la risposta e le variabili indipendenti. La funzione viene calcolata entro dei limiti imposti dall'esperimento. Nel caso sopra citato, la superficie di risposta sarà tridimensionale e visualizzabile. Il punto ottimale corrisponde al minimo della funzione, ovvero quei valori dei parametri che minimizzano l'output di risposta.

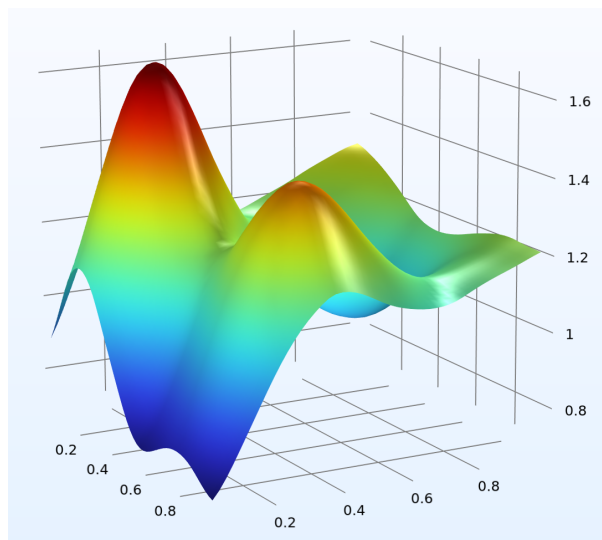


Figura 1.6: Esempio di una superficie di risposta 3D

1.3 Parametrizzazione di forma e mesh morphing

Durante uno studio di ottimizzazione classico devono essere creati più casi di studio per un confronto produttivo. I casi, chiamati anche *Design Points* (DPs), vengono generati attraverso la modifica delle superfici e delle dimensioni in ambiente CAD. Comunemente la generazione della mesh avviene per ogni singolo DP e questa procedura richiede tempo in fase di pre-processing. Per ridurre tali tempi è possibile agire sulla modifica della mesh iniziale operando direttamente sulle posizioni dei nodi della griglia. Questo processo di *mesh morphing* permette di automatizzare l'ottimizzazione e ridurre notevolmente i tempi computazionali, non solo perché creare una nuova mesh è una fase che richiede tempo, ma anche perché ci si avvale dei risultati della soluzione principale per inizializzare i design points, limitando così il tempo necessario per ogni analisi CFD. Nella presente tesi viene proposta una metodologia ibrida in cui si fa uso di CAD parametrici generati da script, i quali vengono confrontati con la geometria di base, chiamata *baseline*, per ottenere un campo di punti e spostamenti. La mesh di

base viene perciò deformata adattandosi alla nuova geometria. Questo metodo assorbe i benefici sia del CAD parametrico (controllo matematico di bordi e superfici) sia del mesh morphing (facilità di automazione e velocità di convergenza).

1.3.1 Il CAD parametrico

Il punto di partenza di ogni analisi fluidodinamica o di altro tipo è sicuramente il disegno tecnico CAD, una rappresentazione dettagliata 2D o 3D di un oggetto. Gli editor CAD più diffusi hanno un'interfaccia grafica attraverso cui il designer modella la geometria. Generalmente, usando questo approccio è difficile definire una serie di parametri ed automatizzare la fase di generazione del CAD. Per questo scopo sono presenti editor CAD con un'interfaccia basata su linguaggi di programmazione differenti. Qui si può definire un numero arbitrario di parametri e vi è la possibilità di automatizzare il processo. Di contro, non essendoci un'interfaccia grafica, il software risulta meno intuitivo e pratico rispetto ai software CAD maggiormente diffusi. Per le analisi riportate in questo elaborato si è fatto uso di due editor CAD che utilizzano "opencascade" come motore CAD. Attraverso la libreria CadQuery disponibile in ambiente Python è stato possibile definire il disegno parametrico del condotto, mentre il disegno parametrico dell'aereo è stato ottenuto con il software open-source Engineering Sketch Pad⁵. Entrambi gli editor sono collegati all'ambiente Python.

1.3.2 Mesh Morphing

Il mesh morphing è una tecnica utilizzata per modificare o deformare una griglia di calcolo mantenendo la topologia inalterata. La modifica tramite CAD semplice non è più necessaria e la mesh non deve essere rigenerata ogni volta. Questa tecnica agisce sulla posizione dei nodi consentendo la modifica anche di solo un insieme di punti.

⁵<https://acdl.mit.edu/ESP/>

La connessione tra le celle viene mantenuta ottenendo così un risultato che si estende per l'intera griglia. Se il metodo si limita al solo riposizionamento dei punti nodali, si potrebbero creare elementi fortemente distorti i quali peggiorano la qualità della mesh. Per evitare questi inconvenienti spesso si effettua un re-meshing delle aree interessate e adiacenti. Questa tecnica è particolarmente efficace quando si lavora con modelli complessi con un numero elevato di componenti e superfici. La riduzione di fasi di lavoro aggiuntive implica una minore possibilità di errori, migliorando così la qualità del lavoro. Sono disponibili due tipologie di algoritmi per il mesh morphing:

- metodi basati sulla mesh;
- metodi basati sul Free Form Deforming (FFD);

Gli algoritmi della prima categoria utilizzano la soluzione di un problema differenziale per aggiornare le celle in base agli spostamenti dei nodi assegnati. Tale tecnica permette di mantenere una qualità elevata e di controllare in modo preciso le superfici. La seconda categoria, invece, utilizza un reticolo di celle cubiche interconnesse il cui spostamento è parametrizzato con delle tecniche polinomiali. Questi metodi si adattano più facilmente a qualsiasi topologia di mesh ma non garantiscono una buona qualità. Un metodo molto valido si basa sull'utilizzo delle Radial Basis Functions (RBF). L'azienda RBF Morph, leader nel mesh morphing, utilizza proprio questo approccio per unire precisione e rapidità di calcolo di un metodo FFD. Il software RBF Morph ⁶ è integrato in Ansys Fluent, un noto solutore commerciale per analisi CFD.

1.3.3 Radial Basis Functions

Dal punto di vista matematico, una funzione radiale di base è una funzione a valori reali il cui valore dipende esclusivamente dalla distanza tra l'argomento della funzione

⁶<https://www.rbf-morph.com/>

e un punto del dominio fissato. Un sistema di queste funzioni può essere utilizzato per approssimare una funzione arbitraria. Dalla definizione di un insieme di punti chiamati "punti sorgente", come si può osservare in figura 1.7, è possibile produrre una soluzione per il loro spostamento. Il metodo è valido per le modifiche di forma sulla superficie

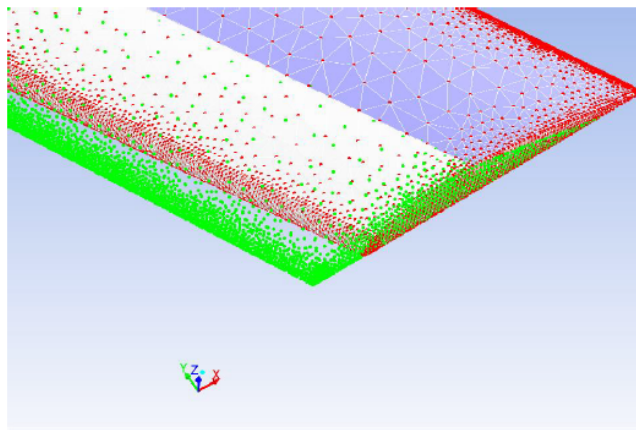


Figura 1.7: Punti sorgente (in rosso) e punti traslati (in verde)

e anche per i volumi. Il metodo RBF rappresenta un mezzo molto potente in grado di interpolare nello spazio una funzione definita in punti discreti fornendone il valore esatto nei punti originali. Esiste una vasta gamma di funzioni RBF: tra le più comuni famiglie di funzioni radiali di base:

- Gaussiane (GS) : $\phi(r) = e^{-r^2}$
- Multi-quadratiche (MQ) : $\phi(r) = \sqrt{1 + r^2}$
- Multi-quadratiche inverse (IMQ) : $\phi(r) = \frac{1}{\sqrt{1 + r^2}}$
- Quadratiche inverse (IQ) : $\phi(r) = \frac{1}{1 + r^2}$
- Spline poliarmoniche : $\phi(r) = r^k$ con $k=1,3,5..$

A seconda della scelta effettuata, si ottiene una diversa interpolazione e differente costo computazionale, consentendo così di ottenere varie combinazioni di complessità e

precisione. Una funzione di interpolazione composta dalle RBF può essere scritta nella seguente forma:

$$s(x) = \sum_{i=1}^N \gamma_i \phi(\|x - x_i\|) + h(x) \quad (1.29)$$

x_i indica le coordinate degli N punti sorgente. La funzione radiale dipende della distanza tra il punto sorgente e il punto finale incognito. I coefficienti γ_i determinano il peso attribuito alla funzione radiale stessa, mentre il termine $h(x)$ rappresenta una correzione polinomiale che garantisce compatibilità con i moti rigidi. I coefficienti e il polinomio $h(x)$ vengono determinati dalle due condizioni al contorno seguenti:

- la funzione $s(x)$ calcolata nei punti sorgente assume il valore degli spostamenti assegnati; $g(x_i)$

$$s(x_i) = g(x_i) \quad (1.30)$$

- i termini polinomiali devono assumere valore nullo nei punti sorgente

$$0 = \sum_{i=1}^N \gamma_i q(x_i) \quad (1.31)$$

Quest'ultima condizione è valida per tutti i polinomi q di ordine minore o uguale del polinomio h . L'ordine minimo di h dipende dal tipo di RBF scelta.

Per ulteriori approfondimenti consultare [1].

Capitolo 2

Strumenti

In questo capitolo vengono illustrati i software utilizzati per effettuare l'ottimizzazione di forma del condotto e dell'aereo.

2.1 CadQuery

CadQuery è una libreria Python intuitiva e pratica per la creazione di modelli CAD 3D parametrici. I principali vantaggi offerti da questo strumento sono:

- L'oggetto viene creato tramite il linguaggio di programmazione Python, un linguaggio standard e ben consolidato
- I modelli parametrici creati possono essere facilmente personalizzati
- Vengono prodotti CAD tridimensionali in formato STEP, AMF e 3MF oltre al tradizionale STL

A titolo di esempio, di seguito viene riportato il codice Python per la creazione di una semplice lastra e il risultato grafico finale in figura 2.1.

```
# importazione della libreria
import cadquery as cq
# definizione delle dimensioni
height = 60.0
width = 80.0
thickness = 10.0
diameter = 22.0 # diametro del foro centrale
padding = 12.0
result = (
    cq.Workplane("XY")
    .box(height, width, thickness)
    .faces(">Z")
    .workplane()
    .hole(diameter)
    .faces(">Z")
    .workplane()
    .rect(height - padding, width - padding, forConstruction=True)
    .vertices()
    .cboreHole(2.4, 4.4, 2.1)
    .edges("|Z")
    .fillet(2.0)
)
# Visualizzazione del solido
show_object(result)

# Esportazione
cq.exporters.export(result, "result.stl")
cq.exporters.export(result.section(), "result.dxf")
cq.exporters.export(result, "result.step")
```

Il comando `cq.Workplane()` identifica il piano di lavoro. Si crea un parallelepipedo con `cq.Workplane.box()` e `cq.Workplane.faces(">Z")` seleziona la faccia superiore nella direzione Z, creando poi un nuovo piano di lavoro con `cq.workplane()`. Il centro del piano di lavoro corrisponde al centro di massa della piastra. `cq.Workplane.hole()` pratica un foro del diametro scelto. Per definire i centri dei 4 fori vicino ai vertici della piastra superiore, viene selezionata la faccia superiore e creato il piano di lavoro. La funzione `cq.Workplane.rect()` disegna un rettangolo con i vertici distanti 6 mm dai bordi del solido. A meno che non venga specificato diversamente, il rettangolo è centrato sul piano di lavoro corrente, in questo caso il centro della faccia superiore della piastra. "forCostruction=True" significa che tale rettangolo non farà

parte del solido, ma che verrà utilizzato per definire altre geometrie. Infatti, dopo aver selezionato i vertici con `cq.Workplane.vertices()`, si creano quattro fori svasati con il comando `cq.Workplane.cboreHole()`. Le dimensioni sono state selezionate per essere compatibili con delle viti a testa cilindrica. Infine, la filettatura viene eseguita tramite `cq.Workplane.fillet()` dopo aver selezionato i bordi paralleli all'asse Z con `cq.Workplane.edges("Z")`. Per concludere, il solido viene esportato attraverso `cq.exporters.export()` in 3 file di tipologia differente.

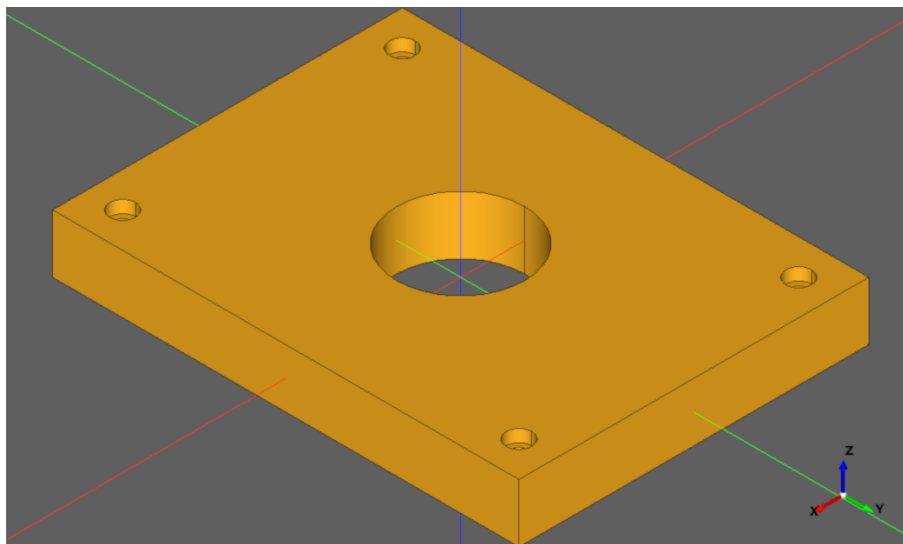


Figura 2.1: Lastra solida ottenuta con CadQuery

2.2 Ansys Workbench

Ansys Workbench è una potente piattaforma di simulazione ingegneristica che ha ottenuto ampi riconoscimenti per la sua versatilità e capacità. Tale strumento offre un ambiente di lavoro unificato per la progettazione e lo sviluppo di una vasta varietà di problemi CAE. Attraverso uno spazio di progettazione dedicato, è possibile gestire le varie interfacce dei prodotti Ansys. In figura 2.2, ad esempio, il prodotto "Ansys Fluent", utilizzato per le simulazioni CFD, è stato selezionato e importato nel progetto.

Lo schema di Fluent è costituito da vari livelli, i quali verranno discussi successivamen-

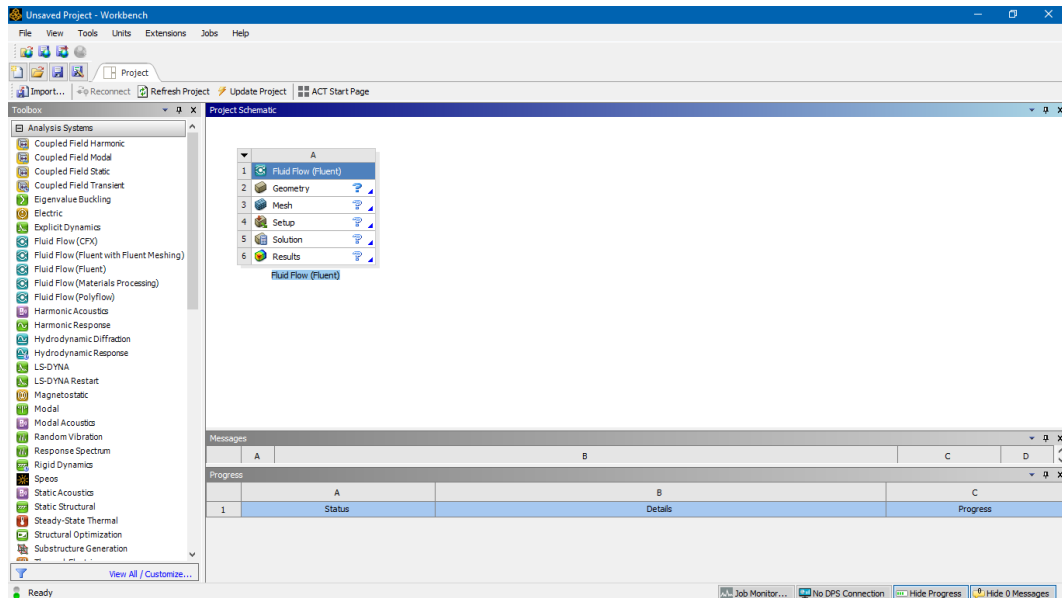


Figura 2.2: Interfaccia Ansys Workbench

te in sezione 2.3. Il grande vantaggio risiede nel poter collegare i vari prodotti tra loro permettendo un'analisi multi-sistema in modo semplice ed intuitivo. Si possono, quindi, effettuare più simulazioni e analisi interdisciplinari. Inoltre, la piattaforma integra funzionalità di post-elaborazione e algoritmi di ottimizzazione. I vantaggi principali sono:

1. Riduzione del tempo di traduzione dei dati ottenuti
2. Strumenti di pre-elaborazione efficaci. Possibilità di importare modelli CAD e di creare griglie di calcolo tramite algoritmi di meshing automatizzati.
3. Ansys Workbench incorpora potenti solutori in grado di simulare accuratamente fenomeni fisici complessi di ogni genere e di fornire informazioni dettagliate sul comportamento dei sistemi in varie condizioni operative.
4. Possibilità di visualizzare e analizzare i risultati della simulazione con un elevato numero di strumenti 2D e 3D

5. Ottimizzazione automatica senza esportazione dei dati

2.3 Ansys Fluent

Il software Ansys Fluent è un solutore CFD della famiglia Ansys in grado di modellare accuratamente flussi turbolenti, trasferimento di calore e reazioni chimiche. Fluent offre un calcolo altamente scalabile e dalle alte prestazioni (High-Performance Computing) per aiutare a risolvere simulazioni di fluidodinamica computazionale complesse e di grandi dimensioni in modo rapido ed economico, sfruttando il calcolo in parallelo. Come già visto in precedenza, il prodotto si suddivide in più fasi, le quali vanno seguite in modo ordinato per una soluzione ottimale.

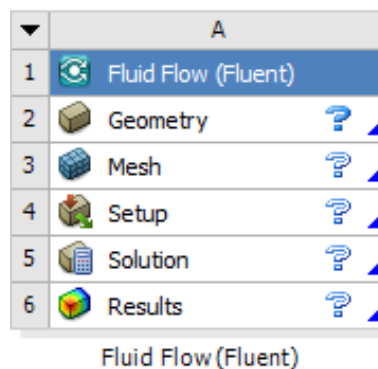


Figura 2.3: Fasi della simulazione CFD

- La geometria può essere importata, creata o modificata attraverso gli strumenti DesignModeler o SpaceClaim. Il primo dei due è più semplice e ottimo per richieste specifiche, il secondo, essendo recente, risulta più potente e versatile.
- La mesh viene generata nell'ambiente "Ansys Fluent Meshing". Si possono realizzare tutte le tipologie di mesh. E' una piattaforma che consente di generare celle volumetriche rapidamente attraverso l'utilizzo di più core (256 al massimo), permettendo di distribuire i processi e ridurre i tempi.

- Per la fase di Set-up si entra effettivamente nell'ambiente di Fluent. Il launcher permette di impostare fondamentalmente il numero di processori, la precisione di macchina e il tipo di risolutore (2D o 3D). L'utilizzo della precisione macchina doppia è necessario per i problemi di fluidodinamica in quanto vicino alla parete le dimensioni delle celle devono essere piccole per garantire un buon studio dello strato limite. Un alto numero di processori permette di ridurre il tempo di soluzione. Infatti, il dominio di calcolo viene diviso in un numero di zone corrispondenti al numero di core. Così facendo si accelera notevolmente la simulazione. Bisogna ricordare che un numero più elevato di partizioni comporta un maggiore tempo di comunicazione tra esse. Una volta avviato Fluent si può osservare la seguente schermata:

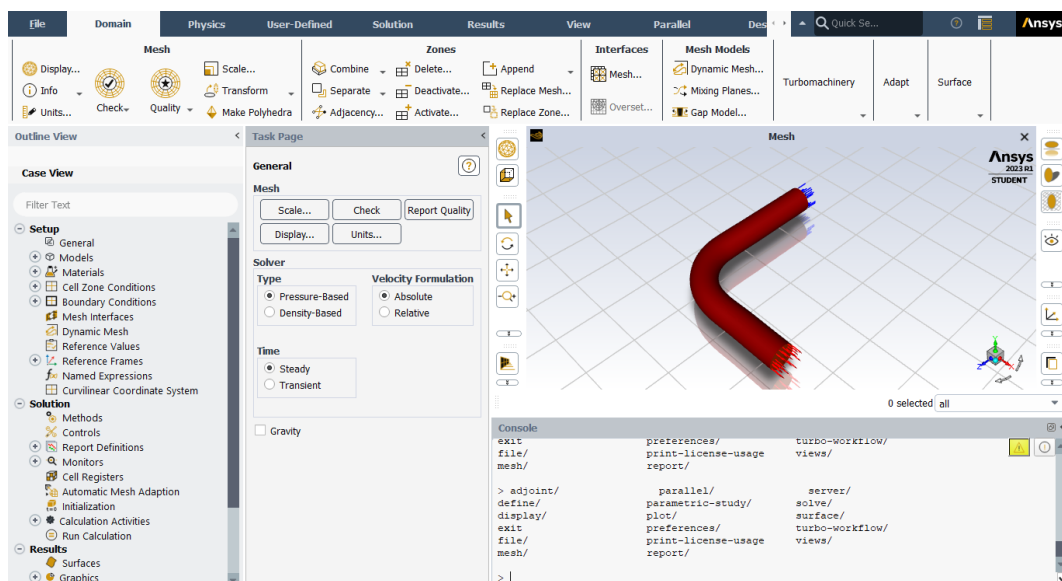


Figura 2.4: Interfaccia di Fluent

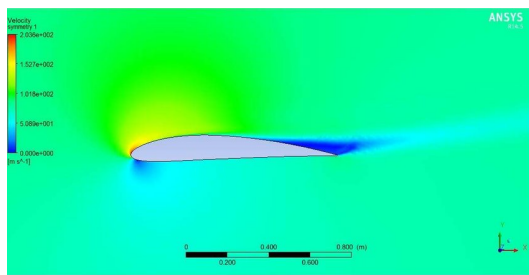
La geometria viene visualizzata nella schermata della Mesh. In basso è presente la console, luogo in cui vengono visualizzati i messaggi provenienti dal solutore. Si possono inoltre scrivere dei comandi diretti per automatizzare e accelerare i processi. A sinistra, il pannello "Outline View" riassume tutte le principali opzioni per la fase di setup, di soluzione e di post-processing. Per la prima fase, tra le impostazioni principali si trova:

- scelta tra simulazione stazionaria o di tipo transitorio;
- scelta tra solutore basato sulla pressione o sulla densità. Il primo viene utilizzato principalmente per flussi incomprimibili o con numero di Mach inferiore a 0.3. Il secondo, invece, è utilizzato per flussi comprimibili;
- scelta di condizioni di simmetria o assialsimmetria;
- scelta del modello di turbolenza;
- scelta del modello di fluido;
- introduzione delle condizioni operative;
- definizione delle condizioni al contorno;

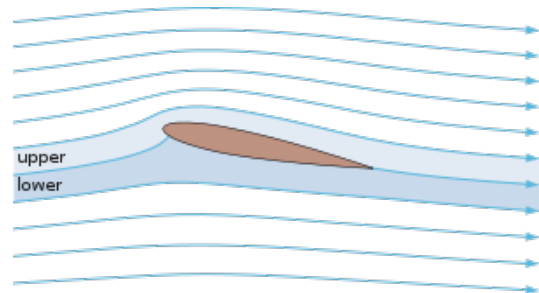
Per la fase di Soluzione:

- scelta tra metodo implicito ed esplicito;
- scelta dell'ordine di accuratezza del metodo;
- nel caso di flussi comprimibili si definisce il numero di Courant;
- scelta dell'ordine di grandezza dei valori residui;
- scelta dei parametri di controllo da monitorare;
- inizializzazione standard o ibrida;
- scelta del numero di iterazioni e simulazione;

Infine, per la fase di post-processing è possibile definire e visualizzare linee di flusso [fig.2.5b], superfici, grafici a contorno [fig.2.5a], report di dati, grafici cartesiani e molto altro. In questa fase è anche possibile calcolare forze e quantità di interesse.



(a) Contour plot di un profilo alare



(b) Linee di flusso di un profilo alare

Figura 2.5: Post-processing

- La fase di soluzione e simulazione si svolge nello stessa piattaforma del setup e le caratteristiche sono state già discusse anteriormente. Di fatto, le fasi di setup e soluzione avvengono in sequenza senza dover cambiare la piattaforma.
- Il post-processing può essere svolto anche in una piattaforma dedicata oltre che in Fluent. Vi è la possibilità di definire nuove variabili e sono presenti tutte le funzionalità elencate in precedenza in Fluent.

2.4 RBF Morph Fluids

RBF Morph Fluids è un mesh morpher perfettamente integrato nella piattaforma di Ansys Fluent vista nella sezione precedente. L'accuratezza dei parametri geometrici e la velocità di deformazione della mesh estremamente rapida rendono questo software adatto a tutti i tipi di problemi fluidodinamici industriali. La topologia della mesh non viene modificata e la sua qualità viene il più possibile preservata. Per modelli di grandi dimensioni in cui il numero di celle è dell'ordine dei milioni, vi è anche la possibilità di utilizzare il calcolo in parallelo.

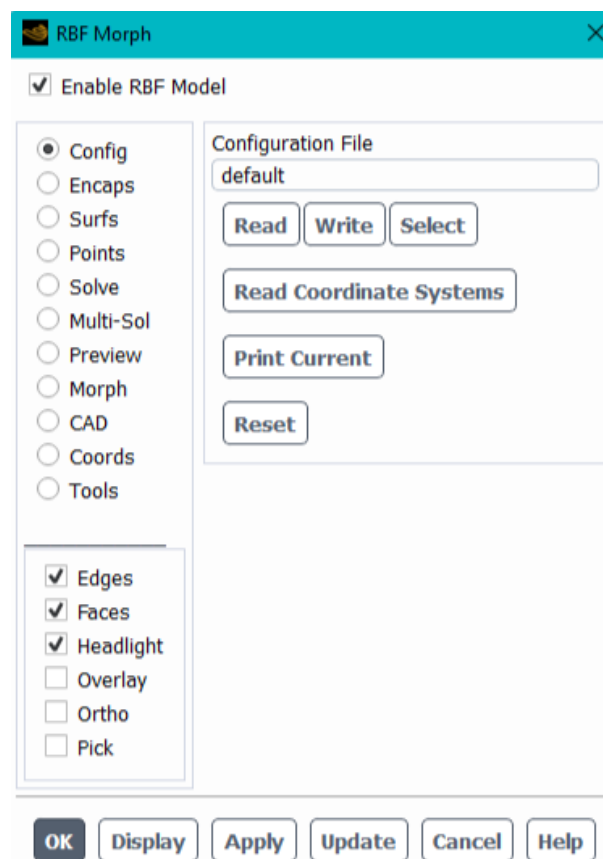


Figura 2.6: Interfaccia di RBF Morph

Per l'impostazione di un caso in RBF Morph Fluids è bene seguire una procedura di tre fasi:

1. Impostazioni generali e definizione dei punti sorgente
2. Soluzione del sistema RBF
3. Modifica della mesh

Lo step 1 richiede l'utilizzo dell'interfaccia grafica [fig.2.6], la quale fornisce tutti gli strumenti necessari per la definizione del problema e dei punti sorgente. In particolare:

- **Config**: si può leggere/scrivere un file di configurazione specificato.
- **Encaps**: la tecnica di incapsulamento viene impiegata per definire i sotto domini del modello all'interno del quale viene applicata l'azione del morpher. Tali domini possono essere definiti mediante vari tipi di forme geometriche (parallelepipedo, cilindro o sfera). Tutto ciò che si trova all'esterno del dominio non è influenzato dalla soluzione RBF. Ogni incapsulamento ha una sua risoluzione, la quale determina il numero di punti equispaziati generati sulla superficie del dominio.
- **Surfs**: in questo pannello è possibile selezionare i punti sorgente dalla superficie della mesh. Per ogni superficie scelta si può attribuire uno spostamento tridimensionale indipendente con il comando "Set M". Lo spostamento può essere anche nullo ed in tal modo si impone la rigidità di quella superficie. E' possibile selezionare tutti i punti di una superficie oppure solamente i punti appartenenti al bordo di essa. Con il comando "Finalize", infine, vengono generati i punti sorgente. I punti possono essere poi visualizzati.
- **Points**: in questa sezione si possono definire punti sorgente singoli attraverso le coordinate e assegnare per ognuno di loro uno specifico spostamento.

Per quanto riguarda lo step 2 della procedura, la soluzione del sistema RBF viene definita e calcolata. Nello specifico:

- **Solve:** nel caso in cui la deformazione desiderata sia simmetrica, si può definire un piano di simmetria in modo da ridurre il costo computazionale. Un'altra utile opzione da considerare è la condizione di periodicità. Tramite il pulsante "Source Points" il sistema colleziona tutti i punti sorgente e, con il tasto "Solution", viene calcolata la soluzione vera e propria. E' possibile leggere e salvare file di soluzione RBF. Infine, si possono visualizzare i punti sia prima che dopo la soluzione.
- **Multi-Sol:** questo pannello permette di combinare più soluzioni assieme con la loro rispettiva e desiderabile amplificazione.

Una volta generata la soluzione si passa alla fase di morphing che include:

- **Preview:** con questa opzione è possibile visualizzare la mesh modificata nell'apposito display senza la modifica effettiva della mesh stessa. Variando il fattore di amplificazione, anche attraverso una sequenza, è possibile valutare al meglio il comportamento dei nodi.
- **Morph:** con il comando "Morph" si modifica la mesh attuale dopo aver selezionato le zone di operatività. Valori troppo elevati di amplificazione possono portare alla generazione di volumi negativi, situazione che provoca il blocco dell'operazione.
- **CAD:** una volta trasformata la mesh e trovata una configurazione di interesse, potrebbe essere necessario produrre un modello CAD di questa configurazione. RBF Morph ha la capacità di produrre un file ".step" ottenuto dalla soluzione RBF.

2.5 Engineering Sketch Pad

Engineering Sketch Pad (ESP¹) è un software di progettazione basato su browser, ideato per creare, modificare e generare modelli solidi tridimensionali utilizzabili in analisi di ottimizzazione di sistemi ingegneristici. Il programma è stato creato utilizzando un'architettura client-server. Il server è costituito da un programma back-end, "serveESP", che esegue la maggior parte del lavoro computazionale da terminale. Il software è compatibile con i più diffusi sistemi operativi. Il client, costruito all'interno di un browser web, fornisce l'interfaccia grafica con la quale l'utente può interagire. ESP è tecnicamente solo l'interfaccia grafica di un sistema di pacchetti software, tra cui:

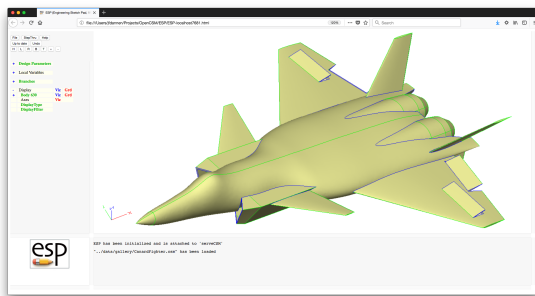
- **WebViewer** è un pacchetto per generare rappresentazioni tridimensionali all'interno di un browser web.
- **OpenCSM** è un modellatore di solidi parametrico. Gli input per OpenCSM sono dei file ".csm", in cui il modello è descritto da una serie di parametri di progettazione e da un albero delle funzionalità. Quest'ultimo è costituito da una serie di comandi standard, operatori booleani e trasformazioni.
- **OpenCASCADE²** è l'unica parte del sistema ESP che deve essere acquisita da una fonte esterna. Open CASCADE Technology (OCCT) è l'unica libreria di geometrie 3D in scala reale open source. E' creata utilizzando il linguaggio di programmazione C++. OCCT è ampiamente utilizzato per lo sviluppo di programmi che si occupano dei seguenti settori ingegneristici e meccanici: modellazione 3D (CAD), produzione (CAM), simulazione numerica (CAE), apparecchiature di misurazione (CMM) e controllo qualità (CAQ).

¹<https://flexcompute.github.io/EngineeringSketchPad/EngSketchPad/ESP/ESP-help.html>

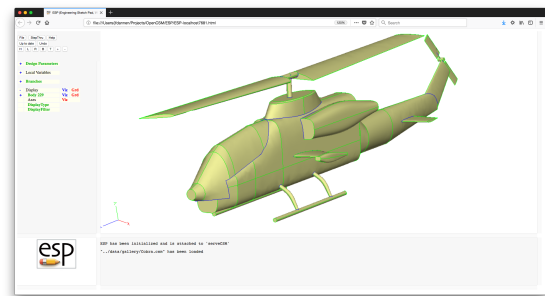
²<https://dev.opencascade.org/>

- **EGADS**, Electronic Geometry Aircraft Design System, è un'interfaccia geometrica open source per OpenCASCADE. Le funzioni al suo interno permettono di configurare in maniera ottimale un sistema aerospaziale.

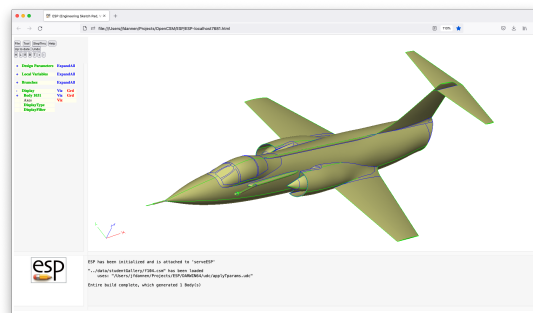
Nelle figure sottostanti si riportano alcune creazioni di esempio.



(a) Canard Fighter



(b) Elicottero



(c) F-104

Figura 2.7: Esempi di sistemi aeronautici ottenuti con ESP

Capitolo 3

Caso di studio: condotto di test

In questo capitolo si riporta il lavoro svolto ed i risultati ottenuti per un condotto di test. La procedura è ordinata e verrà seguita in maniera simile anche per il successivo capitolo in cui si presenterà il caso di studio dell'aereo.

3.1 Generazione del CAD di base

L'oggetto di studio è un condotto ad "L" generato tramite la libreria CadQuery. La configurazione di base, chiamata **baseline**, si genera attraverso il seguente codice:

```
# Importazione delle librerie necessarie
import cadquery as cq
import numpy as np
from cadquery import exporters
# creazione della baseline
s = cq.Workplane("XZ")
# questi sono i punti path della baseline
upPnts = [
    (0.0,0.0),
    (0,50),
    (0,100),
    (0,150),
    (0,200),
    (1,214.1),
    (5,231.22),
    (10,243.59),
```

```
(20,260),
(40,280),
(60,291.65),
(70,295.4),
(80,297.98),
(150,300),
(200,300),
(250,300)
]
# si crea la linea di path interpolando i punti
path = s.spline(upPnts)
# creazione delle aree
# area circolare di ingresso non modificabile
e = cq.Workplane("XY").center(0,0).circle(25)
# area pre curvatura
e2 = cq.Workplane("XY", origin=(0,0,190)).circle(25)
# area post curvatura
e3 = cq.Workplane("ZY", origin=(160,0,300)).circle(25)
# area di uscita non modificabile
e4 = cq.Workplane("ZY", origin=(250,0,300)).circle(25)
# estrusione delle aree con la pathline, creazione del tubo
sweep = cq.Solid.sweep_multi((e.val(),e2.val(),e3.val(),e4.val()),path.val())
# esporto in file .step
exporters.export(sweep, str('C:\\..save_path..\\duct_baseline.step'))
```

Una volta identificato il piano di lavoro XZ, si stabiliscono i punti che formeranno la "path line", ovvero la linea a forma di "L" su cui verrà costruito il condotto. Tale linea viene creata interpolando i punti con il comando `s.spline()`. A questo punto si creano quattro circonferenze di 25 millimetri di raggio: una di ingresso, una di uscita e due agli estremi della curvatura. Infine, `sweep` crea un'estrusione delle sezioni lungo il percorso, ottenendo così il condotto in fig. 3.1. Quest'ultimo viene salvato come file ".step" per essere poi importato in altri ambienti.

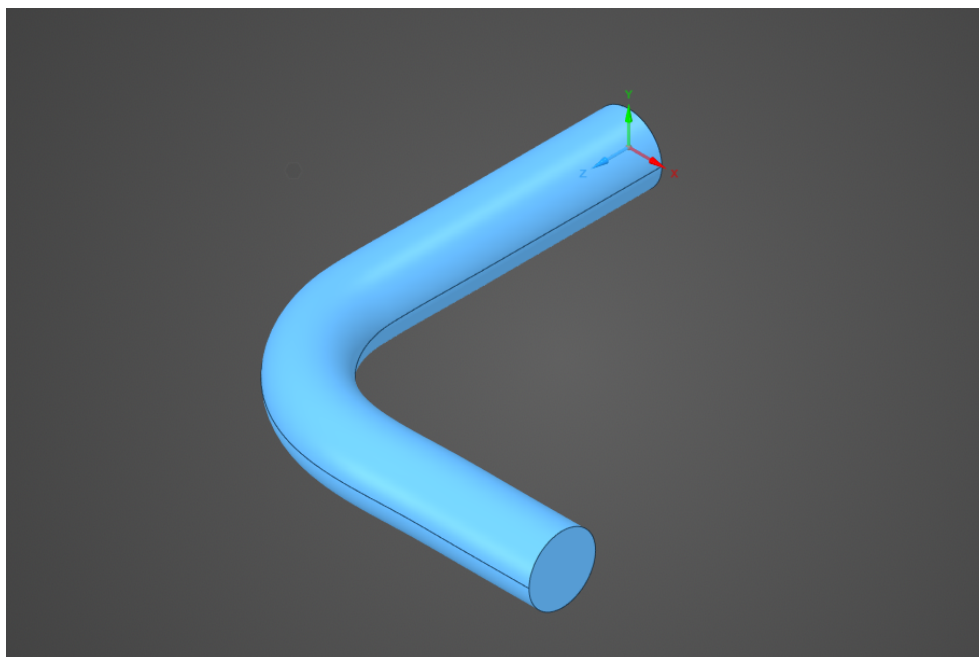


Figura 3.1: CAD baseline del condotto

3.2 Creazione del DOE

Dopo aver creato la baseline, è necessario creare il DOE, impostare i parametri, scegliere il numero di design points che si vuole analizzare e creare un file step per ogni DP. Con riferimento al codice seguente¹:

```
# Importazione delle librerie necessarie
import cadquery as cq
import numpy as np
from scipy.stats import qmc
import pandas as pd
from cadquery import exporters
# Limite inferiore e superiore per i 3 parametri
bounds_l = [-5,23,23] # [Bounds]
bounds_up = [5,27,27] # [Bounds]
n = 3 # numero di parametri
m= 23 # numero di campioni
DP = np.linspace(1,m,m,dtype = int) # array da 1 a 23
```

¹Per chiarezza, nei codici vengono utilizzati i punti di sospensione (...) per indicare il proseguimento alla riga successiva dell'istruzione.

```

# campionatore
sampler = qmc.LatinHypercube(d=n, strength=2, optimization="random-cd")
# matrice campione 23 righe 3 colonne
sample = sampler.random(m)

# valori del campione riscalati tra i limiti
sample = qmc.scale(sample, bounds_l, bounds_up)
# aggiungo limite inf alla riga 24
sample = np.append(sample,bounds_l).reshape(m+1,n)
# aggiungo limite sup alla riga 25
sample = np.append(sample,bounds_up).reshape(m+2,n)

# scrittura in file csv del campione
datapanda = pd.DataFrame(list(zip(DP,sample[:,0],sample[:,1],sample[:,2]))...
,columns = ['DP','path','radius_1','radius_2'])
datapanda.to_csv('points_LHS.csv', index=False)

for i in range (len(sample)):
    s = cq.Workplane("XZ")
    P1 = sample[i,0] # Control of path
    P2 = sample[i,1] # Radius1
    P3 = sample[i,2] # Radius2
    # questi sono i punti path della baseline, in alcuni viene aggiunto P1
    upPnts = [
        (0.0,0.0),
        (0,50),
        (0,100),
        (0,150),
        (0,200),
        (1,214.1+P1),
        (5,231.22+P1),
        (10,243.59+P1),
        (20,260+P1),
        (40,280+P1),
        (60,291.65+P1),
        (70,295.4+P1),
        (80,297.98+P1),
        (150,300),
        (200,300),
        (250,300)
    ]
    path = s.spline(upPnts) # si crea la linea di path interpolando i punti

    e = (cq.Workplane("XY").center(0,0).circle(25))
    # area pre curvatura modificata da P2
    e2 = cq.Workplane("XY", origin=(0,0,190)).circle(P2)
    # area post curvatura modificata da P3
    e3 = cq.Workplane("ZY", origin=(160,0,300)).circle(P3)
    e4 = cq.Workplane("ZY", origin=(250,0,300)).circle(25)

```

```
sweep = cq.Solid.sweep_multi((e.val(),e2.val(),e3.val(),e4.val())...
, path.val())
# esporto in file .step
exporters.export(sweep, str('C:\\..percorso..\\duct' + str(i)+'.step'))
```

Si scelgono tre parametri: il primo (P1) muove alcuni punti della linea di percorso. Il secondo e il terzo, P2 e P3, rappresentano i raggi delle circonferenze in prossimità della curvatura. Per questi tre parametri vengono definiti dei limiti superiori ed inferiori. La variabile **m** rappresenta il numero di campioni. Attraverso la tecnica di campionamento LatinHypercube vengono creati 23 campioni a cui vengono aggiunti i due condotti con i parametri limite, ottenendo un totale di 25 campioni. Per ogni DP vengono trascritti i valori dei parametri in un file ".csv", separati da virgola, come si può notare nella figura 3.2 sottostante:

1	DP,path,radius_1,radius_2
2	1,-3.3692941900029583,23.529480682786463,23.344510559793918
3	2,-0.3203972211525299,23.045431713936853,24.329480682786464
4	3,2.6307058099970417,26.852282323998818,25.929480682786462
5	4,3.6796027788474692,24.071841111538987,26.47184111153899
6	5,-4.138723600515203,24.14451055979392,24.871841111538988
7	6,-2.32039722115253,23.344510559793918,25.129480682786465
8	7,-1.6762982930338408,25.129480682786465,26.54451055979392
9	8,3.1135792848421353,24.452282323998816,23.652282323998815
10	9,-4.32039722115253,24.94451055979392,23.845431713936854
11	10,-2.8864207151578647,24.645431713936855,26.052282323998817
12	11,-0.8864207151578647,24.329480682786464,25.445431713936855
13	12,2.3237017069661583,24.871841111538988,24.645431713936855
14	13,4.630705809997041,25.74451055979392,24.071841111538987
15	14,-4.886420715157865,25.67184111153899,25.74451055979392
16	15,0.3237017069661583,25.445431713936856,25.252282323998816
17	16,1.1135792848421353,25.929480682786462,26.245431713936853
18	17,1.8612763994847974,25.252282323998816,23.045431713936853
19	18,3.8612763994847974,26.245431713936853,24.94451055979392
20	19,-3.6762982930338413,26.47184111153899,26.729480682786463
21	20,-0.1387236005152026,26.54451055979392,23.529480682786463
22	21,-2.1387236005152026,26.729480682786463,24.452282323998816
23	22,-1.3692941900029583,26.052282323998817,23.271841111538986
24	23,4.323701706966158,23.271841111538986,25.67184111153899
25	24,-5,23,23
26	25,5,27,27

Figura 3.2: File csv con tutti i design points

Nella cartella di lavoro vengono prodotti i file step, fig. 3.3

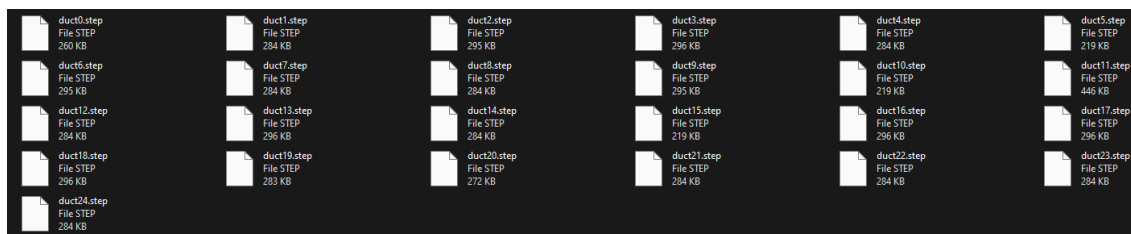


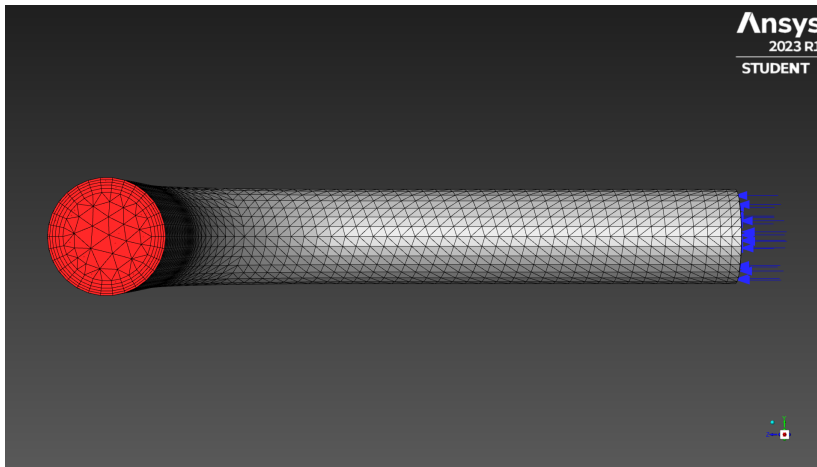
Figura 3.3: Files .step

In conclusione, ogni elemento è un condotto con forma e geometria leggermente modificate rispetto alla configurazione di baseline.

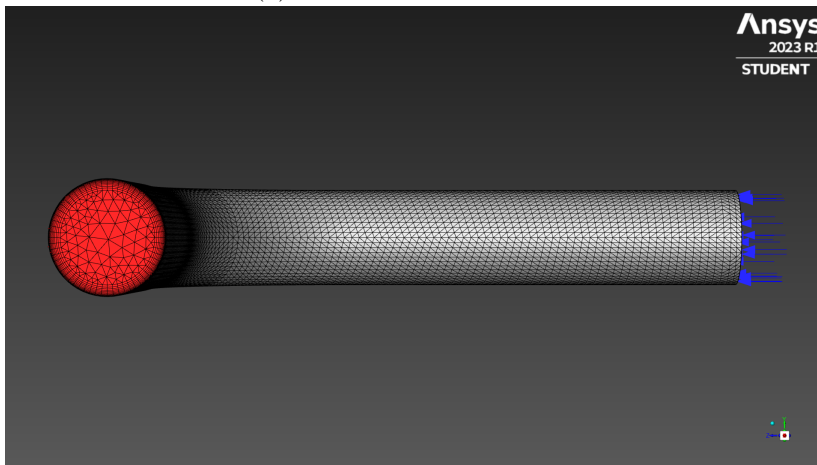
3.3 Creazione della mesh

In ambiente ANSYS Workbench viene importata la geometria di base. Si ricorda che è necessario creare una sola griglia di calcolo per la baseline; per gli altri condotti la mesh sarà il risultato della deformazione della griglia di base tramite RBF. Attraverso ANSYS Meshing si opera per generare la griglia computazionale. Si è deciso di svolgere uno studio di convergenza: in genere una mesh più fine produce una soluzione più accurata. Tuttavia, quando una mesh viene affinata, il tempo necessario per l'elaborazione aumenta. Per questo studio vengono analizzate tre mesh di densità differente. Di base, la mesh è stata generata con questi strumenti di calcolo:

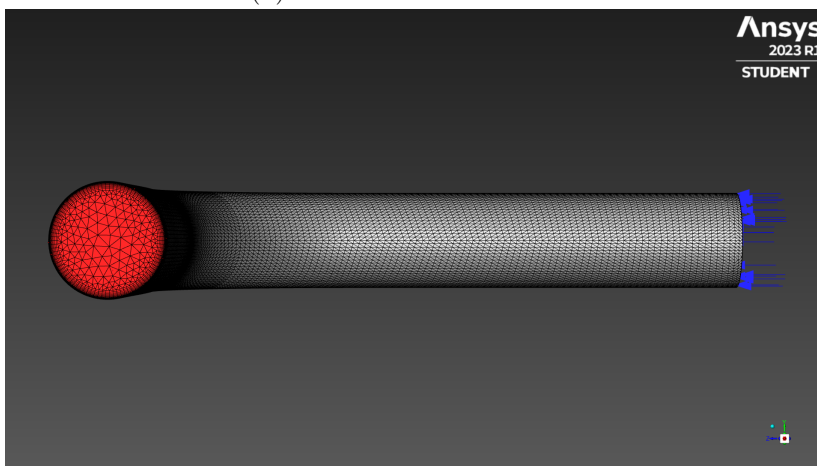
- **Inflation:** progettato per creare elementi sottili in grado di catturare il comportamento vicino alle pareti.
- **Sizing:** comando che permette di cambiare il numero di divisioni o la dimensione degli elementi di un bordo, una superficie o un corpo.
- **Face meshing:** consente di avere mesh di superficie più uniformi e meno distorte.



(a) MESH 1 - densità bassa



(b) MESH 2 - densità media



(c) MESH 3 - densità alta

Figura 3.4: Studio di convergenza

In tabella 3.1 sono riportate le caratteristiche principali delle tre mesh testate, visibili rispettivamente nelle figure 3.4a, 3.4b, 3.4c.

	MESH 1	MESH 2	MESH 3
Densità	Bassa	Media	Alta
Inflation levels	5	10	15
Sizing divisioni circonferenze	15	30	45
Sizing divisioni bordi laterali	60	120	145
Face meshing			
Numero di elementi	31865	220200	470947

Tabella 3.1: Caratteristiche delle griglie per lo studio di convergenza

Per quanto riguarda la qualità, vengono riportati i seguenti dati:

	MESH 1	MESH 2	MESH 3
Max. Skewness	0.79413	0.81924	0.84777
Average Skewness	0.41031	0.4443	0.53023
Max. Aspect Ratio	10.153	37.473	68.257
Max. Orthogonality	0.98188	0.9895	0.98134
Min. Orthogonality	0.20587	0.18076	0.15223
Average Orthogonality	0.58907	0.55549	0.46957

Tabella 3.2: Fattori di qualità

Infine, i valori della velocità massima all'uscita del condotto vengono analizzati per confermare la convergenza della mesh. Si trova:

Max. exit velocity (m/s)	12.6	12.9	12.9
--------------------------	------	------	------

Quindi, effettivamente, con la mesh di densità media si raggiungono già buoni risultati. Per non appesantire troppo la simulazione verrà utilizzata questa griglia (MESH2).

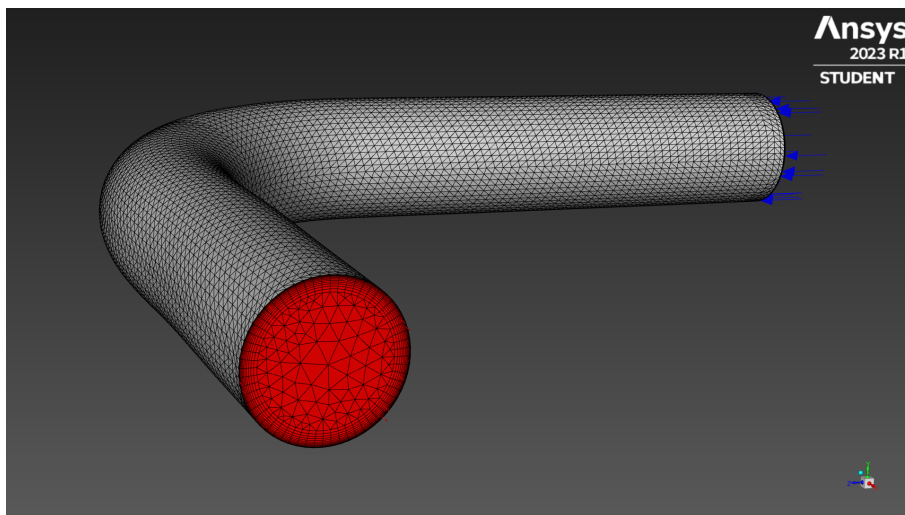


Figura 3.5: Mesh utilizzata per lo studio di ottimizzazione

3.4 Fase di set-up e soluzione

La fase di configurazione del problema si esegue in Fluent. Dopo aver importato il file "MESH2.msh" contenente la griglia stessa, il caso di studio viene impostato attraverso le seguenti opzioni:

- Simulazione stazionaria;
- Solutore pressure-based;
- Modello di turbolenza k-omega SST;
- Aria con densità e viscosità costanti;
- Condizioni di atmosfera standard a livello del mare;
- Condotta di alluminio;
- Condizioni al contorno:
 - **Inlet** [velocity-inlet type]: velocità di ingresso di 10 m/s con un'intensità turbolenta del 5%;
 - **Outlet** [pressure-outlet type]: pressione all'uscita di 101325 Pascal, uguale a quella in ingresso;
- Residual: le equazioni RANS vengono risolte in maniera iterativa. I residui misurano il grado di accuratezza delle variabili osservate (continuità, componenti della velocità, k e omega). Idealmente quando la soluzione converge, questi residui andranno a zero. In un computer reale, i residui diminuiscono fino a raggiungere un valore di minimo per poi stabilizzarsi. Impostiamo un grado di accuratezza di $1e-6$ come obiettivo;

- Parametri di monitoraggio: vengono definiti due parametri fisici, pressione in ingresso (p.in) e velocità media di uscita (v.out), in modo da verificare lo stato della convergenza;
- Metodi di soluzione:

Solution Methods	
Scheme	Coupled
Spatial Discretization	
Gradient	Least Squares Cell Based
Pressure	Second Order
Momentum	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Specific Dissipation Rate	Second Order Upwind
Other Settings	
Pseudo Time Method	Global Time Step
Warped-Face Gradient Correction	Off
High Order Term Relaxation	Off

Tabella 3.3: Metodi di soluzione

- Inizializzazione Standard: condizioni iniziali relative alle condizioni di Inlet;
- Definizione di un osservabile: definiamo la caduta di pressione come un parametro da monitorare e trascrivere alla fine delle simulazioni.

3.5 Risultati baseline

La simulazione ha richiesto meno di 100 iterazioni per raggiungere la convergenza, come si può notare dal grafico in figura 3.6. Tale grafico rappresenta uno dei due parametri scelti per monitorare la convergenza. Il profilo di velocità [fig. 3.7] ottenuto è caratterizzato da zone di velocità diversa.

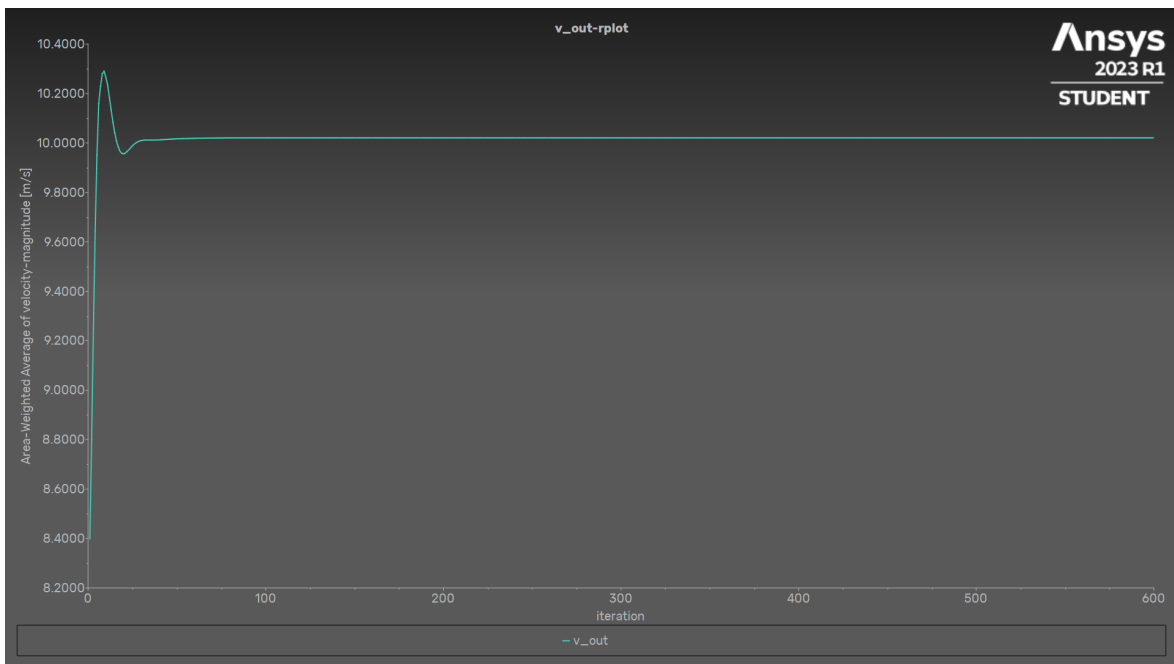


Figura 3.6: grafico di v_out

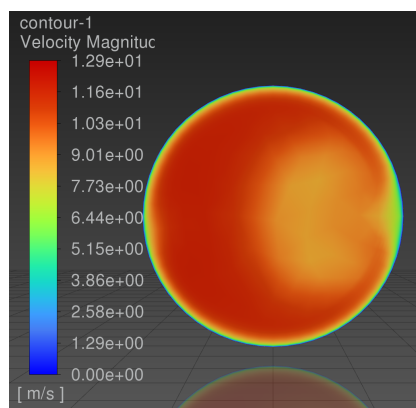


Figura 3.7: Linee di contorno della velocità all'uscita

Vicino alle pareti la velocità si avvicina al valore nullo, mentre la velocità massima si ha in corrispondenza della zona centrale. Mediamente si ha un valore di 10 m/s come ci si aspetta.

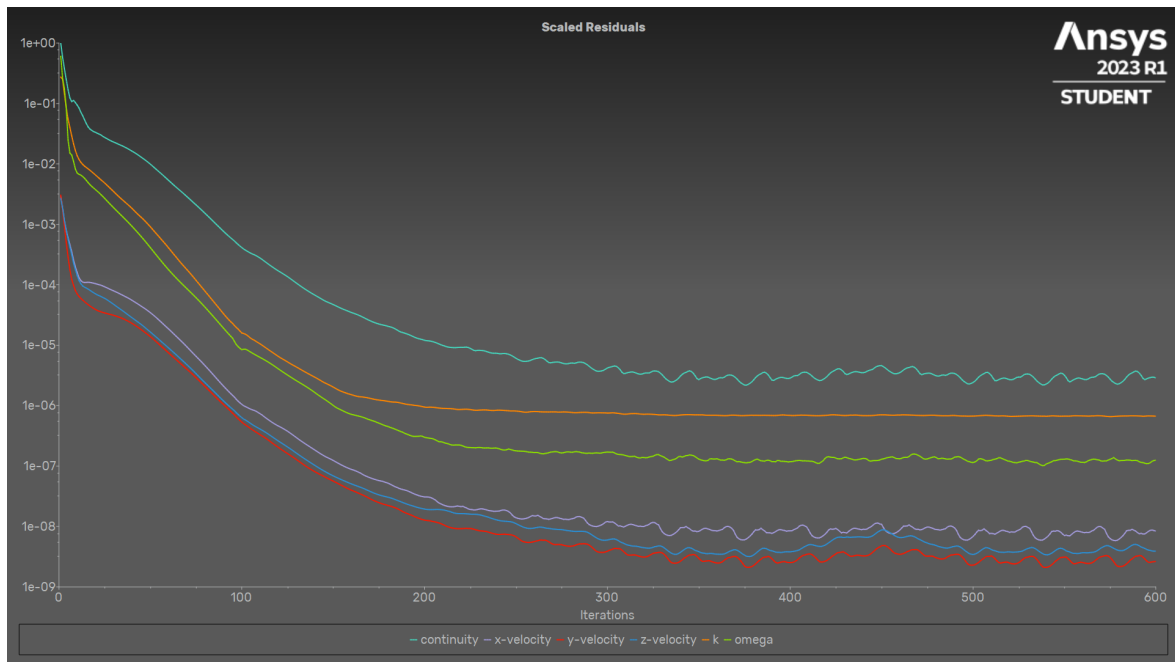


Figura 3.8: Grafico dei residui

Sono state comunque eseguite 600 iterazioni, raggiungendo un grado di accuratezza minimo di $1e-5$ (continuità) e massimo intorno a $1e-8$ (velocità) [fig. 3.8]. La caduta di pressione risulta essere pari a 23.848 Pascal.

3.6 Workflow completo - condotto

Dopo aver ottenuto i risultati della geometria di base, si opera per simulare ogni configurazione modificata. Avendo 25 design points, si utilizza un codice python per automatizzare il processo in cui ogni sequenza di comandi viene ripetuta per ogni DP in un ciclo iterativo "for", comandato dall'indice "i" che ad ogni ciclo aumenta di un'unità.

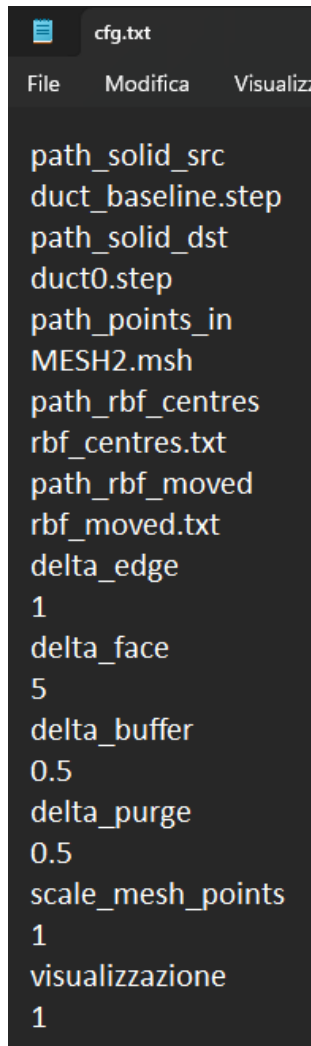
Di seguito si riporta la prima parte del codice dove vengono importate le librerie necessarie, viene letto il file .csv con tutti i campioni e viene salvato il percorso della cartella di lavoro.

```
# Importazione delle librerie necessarie
import pandas as pd
import numpy as np
import os
import shutil
import glob
import subprocess
# Lettura del file CSV in un dataframe "df"
file_path = 'C:\\..percorso del file..\\Mesh2\\points_LHS.csv'
df = pd.read_csv(file_path)
DOE = np.array(df)
nP = len(DOE)
# Salvataggio del percorso della cartella di lavoro iniziale
old_directory = os.getcwd()
```

Ora, si entra nel ciclo iterativo e si eseguono le seguenti operazioni:

- Creazione di una nuova cartella di lavoro per ogni design point.
- In tale cartella vengono copiati tutti i file necessari, tra cui:
 - il file **MESH2.msh** rappresentativo della mesh originale;
 - i file **MESH2.cas.h5** e **MESH2.dat.h5** che rappresentano il caso di studio e i risultati ottenuti in Fluent per la baseline;
 - il CAD in formato "step" della baseline e del design point corrente;

-
- **roc.exe**, un eseguibile programmato in C++ e basato sulla tecnologia Open CASCADE; questo programma genera una densa nuvola di punti in cui vengono trascritte le informazioni della modifica di forma. In particolare, vengono creati due file: `rbf_centres.txt` e `rbf_moved.txt`. Il primo contiene le coordinate dei punti e gli spostamenti nelle tre direzioni causati dalla geometria modificata. Il secondo, in modo simile, contiene le coordinate dei punti dopo gli spostamenti. Per il suo funzionamento, l'eseguibile richiede delle librerie dinamiche (DLL files) specifiche, le quali sono state collegate tramite variabili di sistema;
 - **cfg.txt** [fig. 3.9], un file di testo di input per l'eseguibile `roc.exe`; Nelle prime sei righe vengono definiti i file di input: la baseline (`duct_baseline.step`), la geometria modificata (`duct0.step` per la prima iterazione) e la mesh originale del condotto (`MESH2.msh`). Le quattro righe successive identificano i file di output che si generano dopo l'avvio di `roc.exe`: `rbf_centres.txt` e `rbf_moved.txt`. I tre parametri *delta_edge*, *delta_face* e *delta_buffer* controllano, rispettivamente, il numero di punti sui bordi, il numero di punti sulle facce e lo spazio senza punti tra facce e bordi. In pratica, con questi parametri è possibile controllare manualmente la risoluzione sui bordi e sulle facce, ottenendo una maggiore qualità; *delta_purge* esegue un controllo sui punti duplicati. Le ultime 4 righe sono opzioni di scala e visualizzazione.
 - **duct_surf.rbf**, un file contenente le impostazioni di default di RBF-Morph modificato con questi 4 comandi:
 - (rpsetvar 'rbf/moving-points-from-file-enabled? #t): permette la lettura di punti sorgente da un file esterno;
 - (rpsetvar 'rbf/moving-points-enable-lcm? #t): si attiva la possibilità di creazione del dominio tramite punti dati;

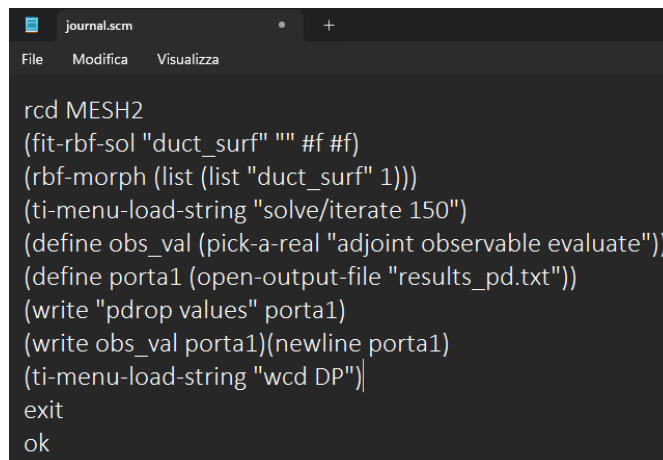


```
cfg.txt
File Modifica Visualizz

path_solid_src
duct_baseline.step
path_solid_dst
duct0.step
path_points_in
MESH2.msh
path_rbf_centres
rbf_centres.txt
path_rbf_moved
rbf_moved.txt
delta_edge
1
delta_face
5
delta_buffer
0.5
delta_purge
0.5
scale_mesh_points
1
visualizzazione
1
```

Figura 3.9: cfg.txt

- (rpsetvar 'rbf/moving-points-filename "mod.pts"):
 - vengono letti i punti e gli spostamenti contenuti nel file mod.pts che verrà introdotto a breve;
 - (rpsetvar 'rbf/moving-points-encap-filter? #t):
 - viene creato il dominio di spostamento.
- **journal.scm** [fig. 3.10], un file di testo ("scheme" con estensione .scm) con una serie di comandi leggibili all'interno di Fluent;
- Attraverso la prima riga viene letto il caso di studio ed i risultati ottenuti per



```

journal.scm
File Modifica Visualizza

rcd MESH2
(fit-rbf-sol "duct_surf" "" #f #f)
(rbf-morph (list (list "duct_surf" 1)))
(ti-menu-load-string "solve/iterate 150")
(define obs_val (pick-a-real "adjoint observable evaluate"))
(define porta1 (open-output-file "results_pd.txt"))
(write "pdrop values" porta1)
(write obs_val porta1)(newline porta1)
(ti-menu-load-string "wcd DP")
exit
ok

```

Figura 3.10: journal.scn

la configurazione di baseline. Le due righe successive consentono la lettura del file "duct_surf.rbf", vengono poi generati i punti sorgenti, viene calcolata la soluzione e la mesh viene deformata. Con il comando (ti-menu-load-string "solve/iterate 150") vengono eseguite altre 150 iterazioni di simulazione per il nuovo condotto utilizzando la convergenza ottenuta con la baseline. Le righe di comando restanti servono a creare un file di testo di output con il valore della caduta di pressione aggiornata. Qui si fa uso del risolutore adjoint, uno strumento di ottimizzazione semiautomatico disponibile in Fluent. Infine, vengono salvati i risultati e viene chiusa l'applicazione (Fluent).

- Ad ogni iterazione è necessario aggiornare il file `cfg.txt` alla riga 4 in modo tale da prendere in considerazione ad ogni ciclo la configurazione modificata corretta.
- Avvio dell'eseguibile **roc.exe** tramite linea di comando del terminale: "roc -mesh `cfg.txt`" è la sintassi corretta per avviare il programma.
- Trasformazione del file `rbf_centres.txt` in **mod.pts** [fig. 3.11], leggibile in RBF-Morph. Nella prima riga viene trascritto il numero totale di punti sorgente. Dalla seconda, ogni riga rappresenta un punto sorgente, le prime 3 colonne rappresenta-

no le coordinate dei punti sorgente (X, Y e Z) e le ultime 3 colonne rappresentano gli spostamenti (DX, DY e DZ).

- Dal terminale, viene infine avviato Fluent senza l'interfaccia grafica (-g) specificando dimensione (3d), numero di processori (-t4) e file journal da leggere (-i journal.scm).

```

mod.pts
File Modifica Visualizza

3980
-0.0176777 -0.0176777 -9.169390000000001e-19 0.0 0.0 1.81261e-18
-0.015 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.014694599999999999 -0.0202254 -9.12247e-19 0.0 0.0 1.86289e-18
-0.0113498 -0.022275200000000002 -8.00122e-19 0.0 0.0 1.74027e-18
-0.0077254199999999999 -0.0237764 -6.27528e-19 0.0 0.0 1.48206e-18
-0.01 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.0075 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.005 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.0125 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.01 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.0075 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.005 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.00391086 -0.0246922 -4.39421e-19 0.0 0.0 1.13788e-18
2.0627899999999997e-15 -0.025 -2.65413e-19 0.0 0.0 7.62411e-19
0.00391086 -0.0246922 -1.14502e-19 0.0 0.0 4.0790599999999997e-19
-0.0025 -0.0225 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.0025 -0.0225 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.0025 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-4.70707e-19 -0.02 3.0826699999999996e-19 1.73717e-18 0.0 1.48706e-19
0.0025 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.005 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.0025 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-4.70707e-19 -0.0175 3.0826699999999996e-19 1.73717e-18 0.0 1.48706e-19
0.0025 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.005 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.0077254199999999999 -0.0237764 2.44287e-20 0.0 0.0 1.1454000000000002e-19
0.0113498 -0.022275200000000002 1.7898e-19 0.0 0.0 -1.0116e-19
0.0146945999999999999 -0.0202254 3.8823e-19 0.0 0.0 -2.57037e-19
0.0075 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.01 -0.02 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.0075 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.01 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.0125 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.015 -0.0175 3.0826699999999996e-19 0.0 0.0 1.48706e-19
0.0176777 -0.0176777 6.96457e-19 0.0 0.0 -4.0952700000000003e-19
-0.0237764 -0.0077254199999999999 -2.5233900000000003e-19 0.0 0.0 6.63208e-19
-0.022275200000000002 -0.0113498 -5.391e-19 0.0 0.0 1.16637e-18
-0.0202254 -0.0146945999999999999 -7.86778e-19 0.0 0.0 1.5738e-18
-0.0175 -0.015 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.015 -0.015 3.0826699999999996e-19 0.0 0.0 1.48706e-19
-0.0175 -0.0125 3.0826699999999996e-19 0.0 0.0 1.48706e-19

```

Figura 3.11: mod.pts

```

for i in range (0,nP):
    dir = str(old_directory + '\DP'+ str(i))
    # creazione di una nuova cartella per ogni DP
    os.mkdir(dir)
    # Apertura della nuova cartella
    os.chdir(dir)
    # Copia i file di interesse
    shutil.copy(str(old_directory + '\MESH2.cas.h5'), 'MESH2.cas.h5')
    shutil.copy(str(old_directory + '\MESH2.dat.h5'), 'MESH2.dat.h5')
    shutil.copy(str(old_directory + '\MESH2.msh'), 'MESH2.msh')
    shutil.copy(str(old_directory + '\duct_baseline.step'), ...
    'duct_baseline.step')
    shutil.copy(str(old_directory + '\duct' + str(i) + '.step'), ...
    str ('duct' + str(i) + '.step'))
    shutil.copy(str(old_directory + '\\roc.exe'), 'roc.exe')
    shutil.copy(str(old_directory + '\cfg.txt'), 'cfg.txt')
    shutil.copy(str(old_directory + '\journal.scm'), 'journal.scm')
    shutil.copy(str(old_directory + '\duct_surf.rbf'), 'duct_surf.rbf')

    # Modifica del file cfg.txt con il nome del file .step corretto
    new_row = str('duct' +str(i) + '.step')
    with open('cfg.txt', 'r') as file:
        lines = file.readlines()
    lines[3] = new_row + '\n'
    with open('cfg.txt', 'w') as file:
        file.writelines(lines)

    # Avvio dell'eseguibile per generare la nuvola di punti
    command = 'roc -mesh cfg.txt'
    subprocess.run(command, shell=False, check=True)

    # Creazione del file mod.pts leggibile in RBF-Morph da rbf_centres.txt
    fileIn = "rbf_centres.txt"
    with open(fileIn) as ifile:
        lines = (ifile.readlines())
    npts = len(lines)
    X1 = [0]*npts
    Y1 = [0]*npts
    Z1 = [0]*npts
    dX1 = [0]*npts
    dY1 = [0]*npts
    dZ1 = [0]*npts

    for i in range (0, npts):
        a,b,c,d,e,f =(lines[i].split())

        dX1[i] = (float(d))/1000
        dY1[i] = (float(e))/1000
        dZ1[i] = (float(f))/1000

```

```
X1[i] = (float(a))/1000
Y1[i] = (float(b))/1000
Z1[i] = (float(c))/1000

lines[i] = ".join(str(' '+str(X1[i]) + ' ' + str(Y1[i]) +...
' ' + str(Z1[i]) + ' ' + str(dX1[i]) + ' ' +...
str(dY1[i]) + ' ' + str(dZ1[i]) + '\n'))

# Scrittura del file mod.pts
with open("mod.pts", 'w') as f:
    f.write(str(npts)+'\n')
    for u in range (1,npts) :
        f.write(str(lines[u-1]))
f.close ()

# Avvio di Fluent
command2 = r"D:\..percorso..\fluent" 3d -t4 -g -i journal.scm &'
subprocess.run(command2, shell=True, check=True)
# Spostamento nella cartella di lavoro iniziale
os.chdir(old_directory)
```

3.7 Condotta di esempio

Per comprendere al meglio il processo, si riporta un condotto di esempio. Si considera il condotto modificato “duct10.step”, in cui le aree in prossimità della curvatura sono leggermente ridotte rispetto alla configurazione di base [fig 3.12]. Il codice “run_fluent.py” esegue il ciclo di operazioni in base al valore della variabile di iterazione “i”. Nel momento in cui si avrà $i=10$, il processo verrà eseguito per il condotto di esempio.

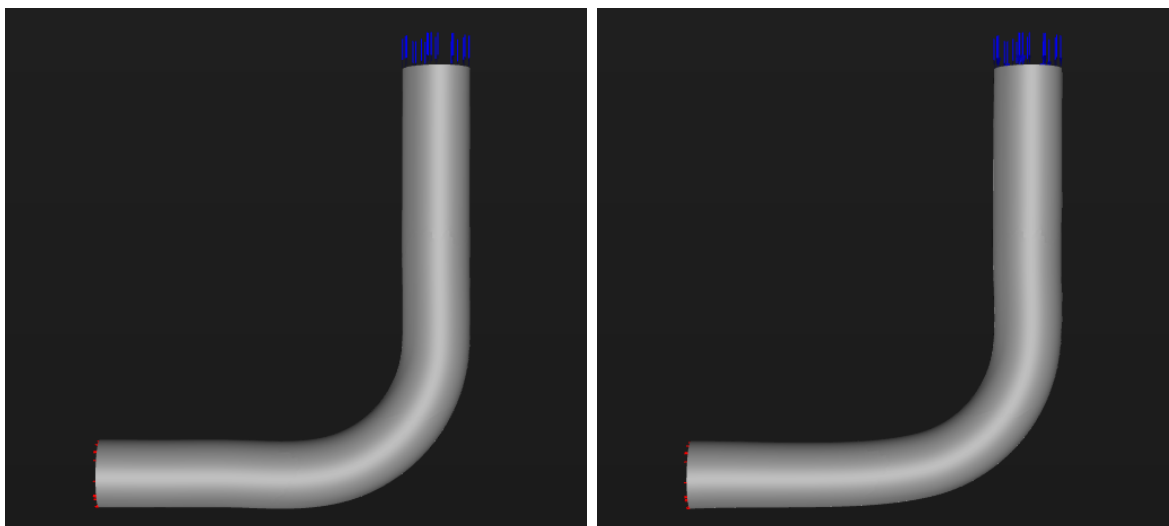


Figura 3.12: Geometrie a confronto: a sinistra la baseline, a destra il DP in questione

Inizialmente viene creata la cartella DP10 in cui vengono trasferiti tutti i file necessari già analizzati nella sezione precedente. Nel file `cfg.txt` viene modificata la riga quattro con “duct10.step”. Avviando l’eseguibile `roc.exe` viene generata la nuvola di punti. Le coordinate dei punti e degli spostamenti calcolati sono espresse in millimetri. Ogni valore viene diviso per 1000 in modo tale da avere l’unità di misura corretta (in metri) in RBF-Morph e in Fluent quando viene letto il file `mod.pts`.

Nelle due figure seguenti si può osservare la nuvola di punti iniziale (in rosso), ovvero i punti sorgente, e i punti finali (in verde) calcolati tramite le funzioni RBF.

Nella seconda immagine si evidenziano più chiaramente le zone che si modificano in corrispondenza della curvatura.

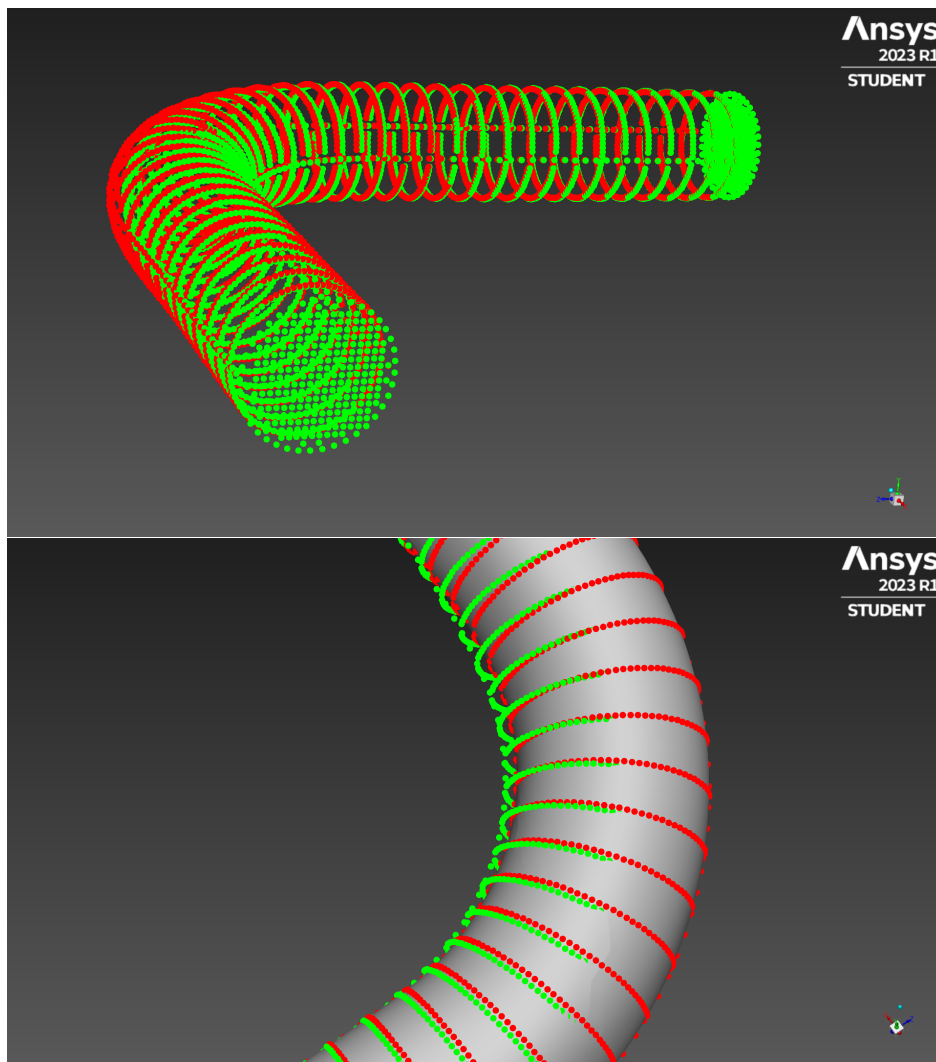


Figura 3.13: punti sorgente e punti finali

Una volta eseguito il mesh morphing, si ha:

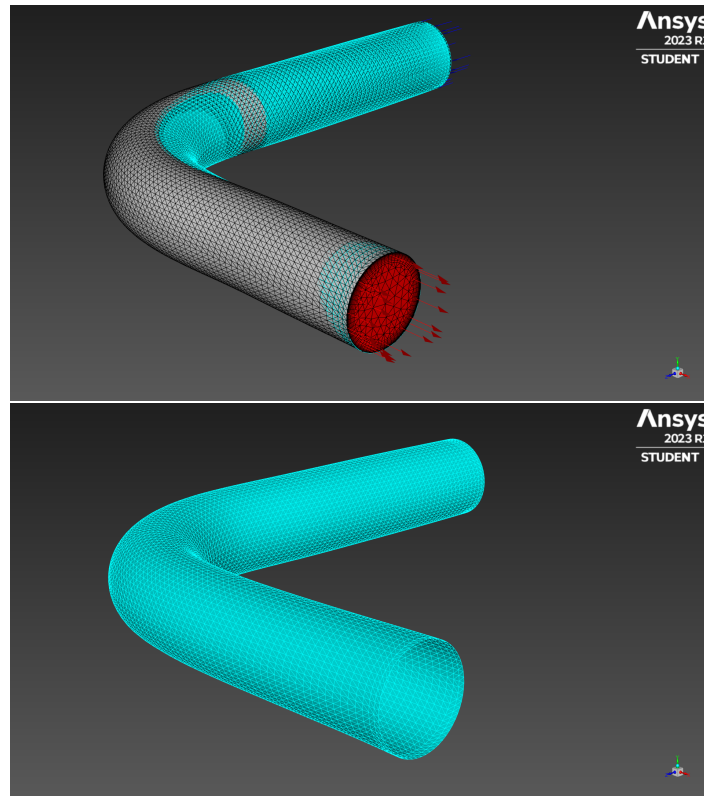


Figura 3.14: Confronto tra mesh modificata (in azzurro) e mesh originale

Dopo 150 iterazioni, il valore di caduta di pressione ottenuto in questo caso è di 25.289101 Pascal, quindi maggiore rispetto alla configurazione di baseline. La mesh modificata risulta avere una qualità pressoché identica a quella originale. Questo è dovuto anche al fatto che gli spostamenti sono contenuti. Infatti, più gli spostamenti sono marcati, più la qualità della mesh tende a peggiorare. In altre parole, una deformazione maggiore della griglia comporta una maggiore riduzione della qualità.

3.8 Ottimizzazione

Alla fine del processo iterativo si ottiene un valore diverso di perdita di pressione totale per ogni design point. Attraverso un ulteriore codice in python [3.1] vengono inseriti

tali risultati nel file "csv" già visto in figura 3.2, aggiungendo la colonna "pressure drop", fig. 3.16a.

Codice 3.1: creazione del file csv aggiornato

```
import pandas as pd
import numpy as np
import os

n = 3 # number of parameters
m= 25 # number of samples
DP = np.linspace(1,m,m,dtype = int) # array da 1 a 25
PD = np.zeros(m)

file_path = 'C:\\\\.percorso del file\\.points_LHS.csv'
df = pd.read_csv(file_path)
old_directory = os.getcwd()
for i in range (0,25) :
    dir = str(old_directory + '\\DP'+ str(i))
    os.chdir(dir)
    with open('results_pd.txt', 'r') as file:
        lines = file.readlines()
        x=lines[0]
        PD[i] = (x[14:23])
    os.chdir(old_directory)

datapanda = pd.DataFrame(list(zip(DP,df['path'],df['radius_1'],...
df['radius_2'],PD)), columns = ['DP','path','radius_1','radius_2'...
,'Pressure drop'])
datapanda.to_csv('results.csv', index=False)
```

Nell'ambiente di ottimizzazione del Workbench [fig. 3.15] è possibile importare il file contenente i parametri ed i risultati corrispondenti come si può notare in figura 3.16b.

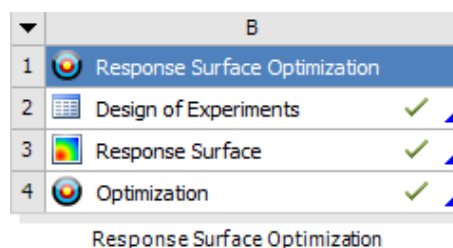


Figura 3.15: Strumento di Ottimizzazione

1	DP,path,radius_1,radius_2,Pressure drop
2	1,-3.3692941900029583,23.529480682786463,23.344510559793918,30.82523
3	2,-0.3203972211525299,23.045431713936853,24.329480682786464,31.512394
4	3,2.6307058099970417,26.852282323998818,25.929480682786462,19.132881
5	4,3.6796027788474692,24.071841111538987,26.47184111153899,27.874138
6	5,-4.138723600515203,24.14451055979392,24.871841111538988,26.06525
7	6,-2.32039722115253,23.344510559793918,25.129480682786465,28.772717
8	7,-1.6762982930338408,25.129480682786465,26.54451055979392,22.090542
9	8,3.1135792848421353,24.452282323998816,23.652282323998815,28.2682
10	9,-4.32039722115253,24.94451055979392,23.845431713936854,25.289101
11	10,-2.8864207151578647,24.645431713936855,26.052282323998817,23.651009
12	11,-0.8864207151578647,24.329480682786464,25.445431713936856,25.247833
13	12,2.3237017069661583,24.871841111538988,24.645431713936855,25.039108
14	13,4.630705809997041,25.74451055979392,24.071841111538987,24.24987
15	14,-4.886420715157865,25.67184111153899,25.74451055979392,21.012356
16	15,0.3237017069661583,25.445431713936856,25.252282323998816,22.464558
17	16,1.1135792848421353,25.929480682786462,26.245431713936853,20.530907
18	17,1.8612763994847974,25.252282323998816,23.045431713936853,27.236675
19	18,3.8612763994847974,26.245431713936853,24.94451055979392,21.593559
20	19,-3.6762982930338413,26.47184111153899,26.729480682786463,18.728085
21	20,-0.1387236005152026,26.54451055979392,23.529480682786463,22.93565
22	21,-2.1387236005152026,26.729480682786463,24.452282323998816,20.649765
23	22,-1.3692941900029583,26.052282323998817,23.271841111538986,24.337307
24	23,4.323701706966158,23.271841111538986,25.67184111153899,30.224892
25	24,-5,23,23,33.581978
26	25,5,27,27,18.296768

(a) File csv dei risultati

Outline of Schematic B2: Design of Experiments

	A	B
1		Enabled
2	Design of Experiments	
3	Input Parameters	
4	P1 - Path	<input checked="" type="checkbox"/>
5	P2 - Radius 1	<input checked="" type="checkbox"/>
6	P3 - Radius 2	<input checked="" type="checkbox"/>
7	Output Parameters	
8	P4 - PD	
9	Charts	
10	Parameters Parallel	<input checked="" type="checkbox"/>
11	Design Points vs Parameter	<input checked="" type="checkbox"/>

Table of Outline A4: Design Points of Design of Experiments

	A	B	C	D	E
	Name	P1 - Path	P2 - Radius 1	P3 - Radius 2	P4 - PD
1	1	-3,3693	23,529	23,345	30,825
2	2	-0,3204	23,045	24,329	31,512
3	3	2,6307	26,852	25,929	19,133
4	4	1,6796	23,652	26,852	27,845
5	5	-4,1387	24,145	24,872	26,065
6	6	0,63071	23,845	24,145	28,773
7	7	-1,6763	25,129	26,545	22,091
8	8	3,1136	24,452	23,652	28,268
9	9	-4,3204	24,945	23,845	25,289
10	10	-2,8864	24,645	26,052	23,651
11	11	-0,88642	24,329	25,445	25,248
12	12	2,3237	24,872	24,645	25,039
13	13	4,6307	25,745	24,072	24,25
14	14	-4,8864	25,672	25,745	21,012
15	15	0,3237	25,445	25,252	22,465
16	16	1,1136	25,929	26,245	20,531
17	17	1,8613	25,252	23,045	27,237
18	18	3,8613	26,245	24,945	21,594
19	19	-3,6763	26,472	26,729	18,728
20	20	-0,13872	26,545	23,529	22,936
21	21	-2,1387	26,729	24,452	20,65
22	22	-1,3693	26,052	23,272	24,337
23	23	4,3237	23,272	25,672	30,225
24	24	-5	23	23	33,582
25	25	5	27	27	18,297
*	New Design Point				

Properties of Outline A8: P4 - PD

	A	B
1	Property	Value
2	General	
3	Units	
4	Values	
5	Calculated Minimum	18,297
6	Calculated Maximum	33,582

(b) Importazione del DOE in Workbench

Il passo successivo consiste nella generazione delle superfici di risposta. Si è scelto di utilizzare superfici di risposta di tipo "Genetic Aggregation" e si ottengono le superfici tridimensionali distinte tra i vari parametri di input, visibili in figura 3.17.

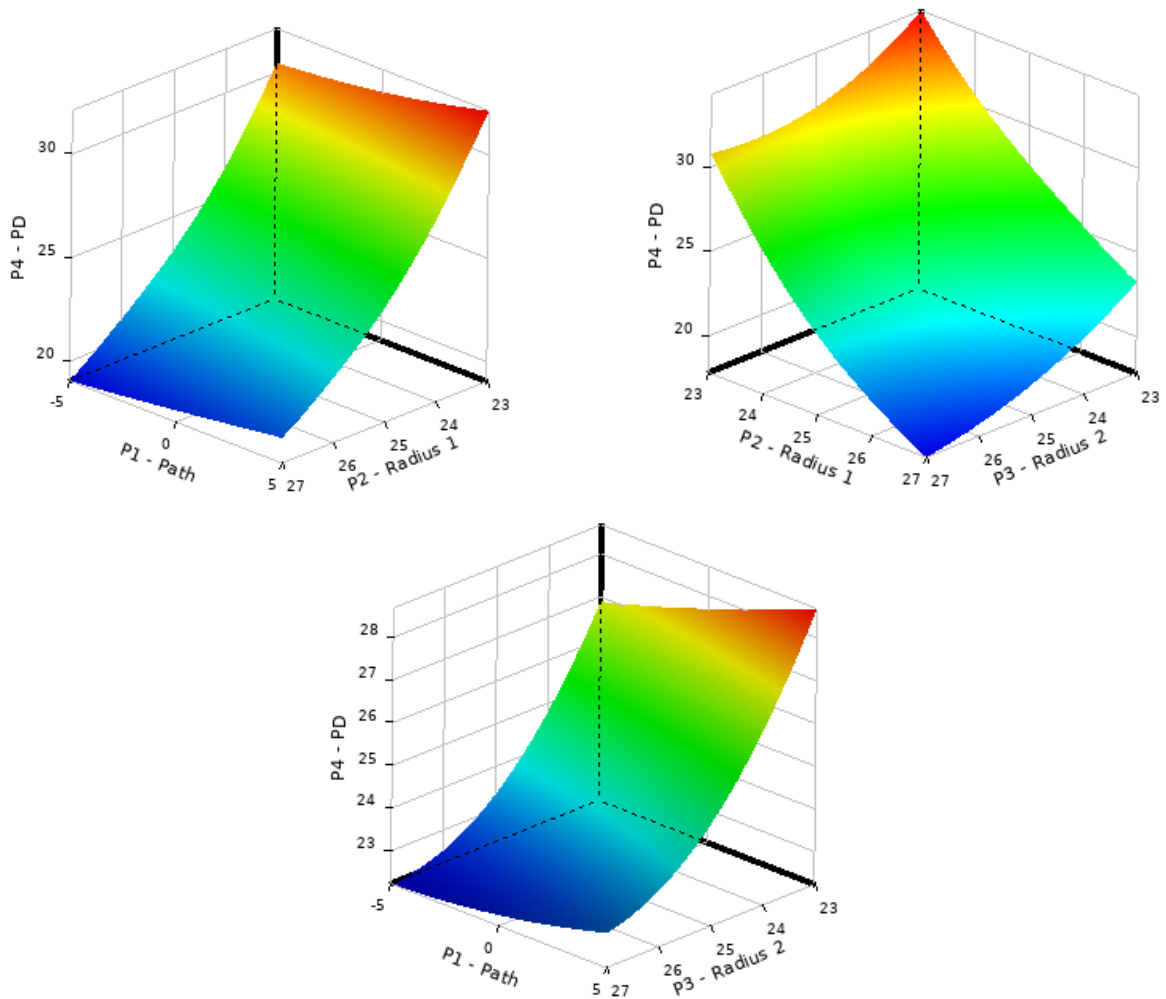


Figura 3.17: Superfici di risposta del condotto

MOGA è il metodo di ottimizzazione che è stato utilizzato. L'obiettivo è minimizzare la caduta di pressione:

Table of Schematic B4: Optimization									
	A	B	C	D	E	F	G	H	I
1	Name	Parameter	Objective			Constraint			
2			Type	Target	Tolerance	Type	Lower Bound	Upper Bound	Tolerance
3	Minimize P4	P4 - PD	Minimize ▾	0		No Constraint ▾			
*		Select a Parameter ▾							

Figura 3.18: Impostazione di ottimizzazione

Dopo che viene eseguita l'ottimizzazione, è possibile visualizzare tre punti candidati [fig. 3.19], i quali identificano le migliori combinazioni per raggiungere lo scopo richiesto.

Table of Schematic B4: Optimization , Candidate Points							
	A	B	C	D	E	F	G
1	Reference	Name ▾	P1 - Path ▾	P2 - Radius 1 ▾	P3 - Radius 2 ▾	P4 - PD ▾	
2						Parameter Value	Variation from Reference
3	☉	Candidate Point 1	-4,1944	26,973	26,911	== 17,592	0,00%
4	☉	Candidate Point 2	-2,131	26,976	26,934	== 17,723	0,74%
5	☉	Candidate Point 3	-2,5437	26,95	26,937	== 17,738	0,83%
*		New Custom Candidate Point	0	25	25		

Figura 3.19: Punti candidati

Inoltre, è possibile visualizzare la sensibilità del parametro di uscita in base ai vari parametri di ingresso [fig. 3.20], ovvero comprendere quale parametro influisce maggiormente sulle prestazioni. Dal grafico si deduce che il parametro più rilevante è la dimensione dell'area dopo la curvatura. Il parametro P1 tende al valore di -5 che rappresenta un raggio di curvatura minore.

E' importante verificare che la configurazione ottimale calcolata nell'ambiente di ottimizzazione rispecchi effettivamente il comportamento fluidodinamico previsto. A tal fine, dal primo punto candidato, viene creato un modello di condotto denominato

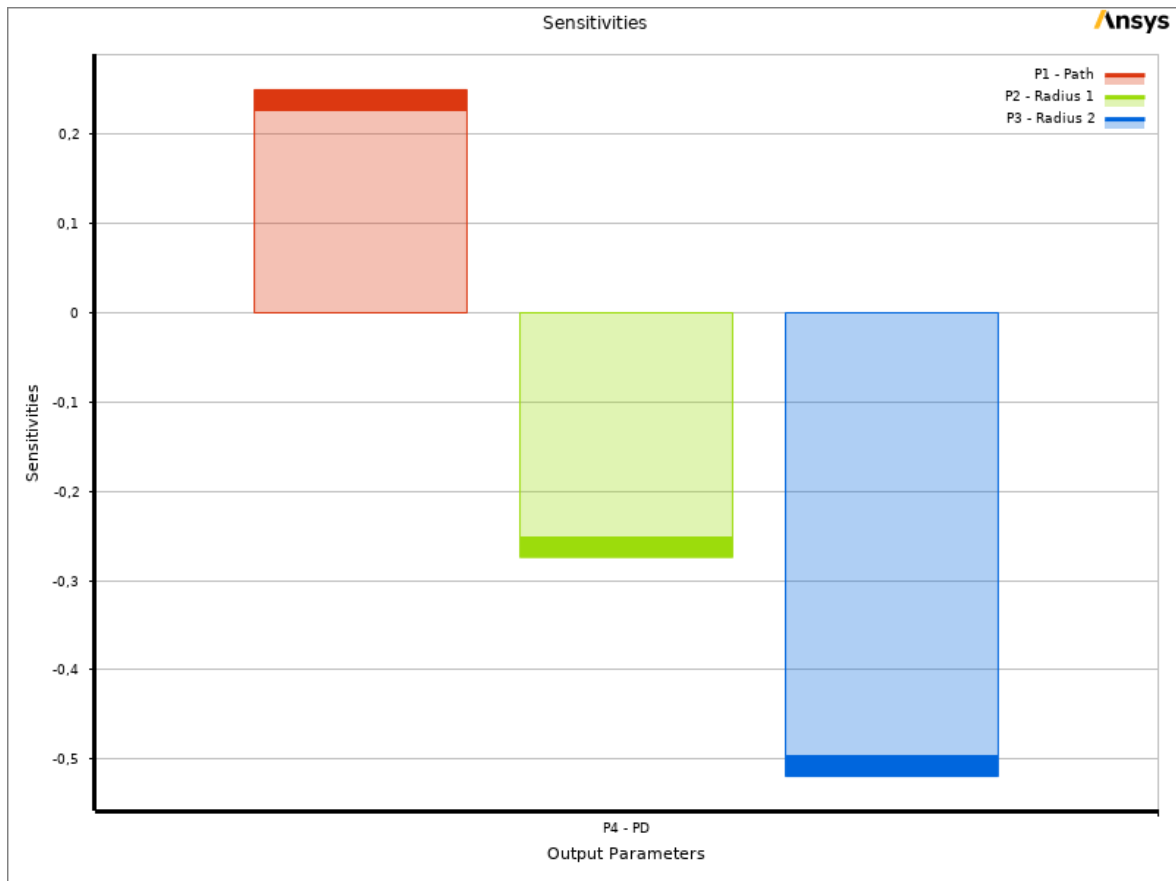


Figura 3.20: Sensibilità locale

”ductOPT.step” impostando i parametri P1, P2 e P3 visibili in figura 3.19. Ripetendo tutta la procedura già illustrata, si ottiene un valore di caduta di pressione pari a 17.681 Pascal. Questo risultato si può ritenere affidabile e vicino al risultato ipotizzato dall’algoritmo di 17.592 Pa. Così facendo la caduta di pressione è stata ridotta del 26.23%.

Capitolo 4

Caso di studio: aeromobile civile

Il 5 e il 6 gennaio 2019 a San Diego in California si è tenuta la seconda edizione di un seminario organizzato dall'American Institute of Aeronautics and Astronautics (AIAA), ente privato americano che opera nel settore ingegneristico aerospaziale da oltre 50 anni. In questi eventi, il terzo e ultimo nel 2022, è stato valutato lo stato dell'arte nella geometria e nelle tecnologie software per la generazione di mesh applicate a sistemi aerospaziali. Il progetto GMGW (Geometry and Mesh Generation Workshops) aveva lo scopo di identificare opportuni miglioramenti e buone pratiche per una generazione di mesh maggiormente precisa ed affidabile. Grazie alla partecipazione di varie aziende specializzate nel settore, tra cui anche RBF Morph in collaborazione con Ansys, sono state condivise idee e soluzioni innovative.

4.1 Configurazione base del velivolo

Per il secondo incontro GMGW, il Dr. John Dannenhoffer, sviluppatore principale del software Engineering Sketch Pad, ha fornito un modello parametrico di aereo (Open Parametric Aircraft Model – OPAM) ispirato al Boeing 787-800 Dreamliner, un velivolo civile bimotore ampiamente utilizzato per tratte di medio-lungo raggio.

Il modello semplificato [fig. 4.1] è basato su 53 parametri di progetto modificabili. Le dimensioni sono in millimetri e l'aereo risulta scalato di circa 300 volte rispetto alle dimensioni reali.

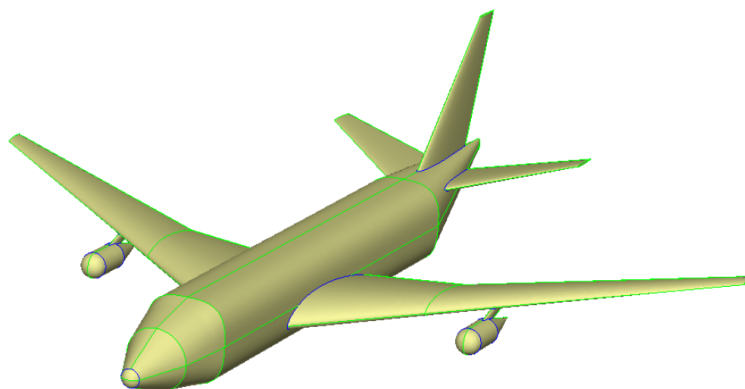


Figura 4.1: Modello di Boeing 787-800

Il disegno tecnico viene generato da un file con estensione “.csm”, il quale viene interpretato da ESP, riprodotto graficamente e salvato come file step. Per semplificazione, in questo elaborato vengono rimossi i motori ed i relativi piloni alari, ottenendo l'aeromobile visibile in figura 4.2.

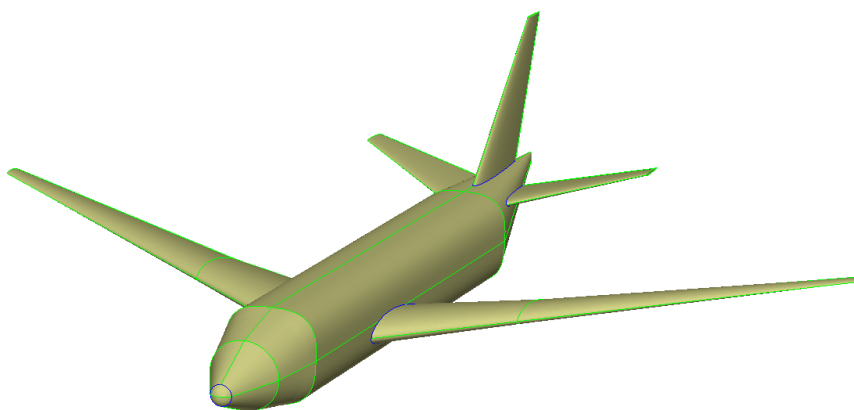


Figura 4.2: Modello modificato

I motori, oltre a non essere di dimensioni reali, non forniscono alcuna spinta, perciò la loro assenza comporterà solamente un valore della forza di resistenza aerodinamica inferiore. Per rendere il modello più realistico possibile è stato impostato un angolo di incidenza alla radice dell'ala di 5° con uno svergolamento della stessa di 2° a circa metà ala e di -1° all'estremità.

In tabella 4.1 vengono riportate le principali caratteristiche di progettazione.

	Modello in scala	B-788
Lunghezza totale	189.5 mm	56.72 m
Larghezza fusoliera	20 mm	5.77 m
Apertura alare	195.35 mm	60.12 m
Superficie alare	4240 mm ²	377 m ²
Angolo di freccia alare	32.2°	

Peso operativo a vuoto	119 950 kg
Peso massimo senza carburante	161 025 kg
Peso massimo al decollo (MTOW)	227 930 kg

Tabella 4.1: Caratteristiche del modello e del velivolo reale

Per avere un'idea generale del modo in cui l'aeromobile viene generato, si riporta in Appendice B il file "baseline_mod.csm" nel quale, dopo aver definito tutti i parametri, vengono generate le varie componenti. Di fatto, in questo elaborato, si presta maggiore attenzione al modo in cui vengono modificati i parametri e non sull'effettiva procedura di creazione dell'oggetto.

4.2 Creazione del DOE

La tecnica di campionamento (LHS) utilizzata in questo caso di studio è la medesima di quella utilizzata per il condotto di prova. Si è scelto di utilizzare 5 parametri di riferimento:

- 1) Allungamento alare dell'ala con superficie fissata, indicato con il parametro P1 - wing:aspect
- 2) Rapporto tra lo spessore del profilo e la corda alla radice dell'ala: P2 - wing:thickr
- 3) Posizione della radice delle ali in direzione longitudinale, attraverso il parametro P3 - wing:xroot
- 4) Angolo di incidenza del profilo in corrispondenza della sezione che divide l'ala tra zona esterna ed interna, indicato con P4 - wing:alphab
- 5) potenza della superellisse della fusoliera centrale, con il parametro P5 - fuse:power

I 5 parametri sono stati fatti variare entro dei limiti imposti, visibili nella tabella 4.2 sottostante.

Parametro		Valore nominale	Limite inferiore	Limite superiore
wing:aspect	P1	9	7	11
wing:thickr	P2	0.15	0.1	0.2
wing:xroot	P3	54	51	57
wing:alphab	P4	2°	1°	3°
fuse:power	P5	3	2	4

Tabella 4.2: Range di valori utilizzati

Si è deciso di scegliere un numero di campioni pari a 50, ai quali vengono aggiunti due campioni aventi i parametri limite. Come per il condotto, anche in questo caso

viene generato un file CSV, denominato "points_LHS_plane.csv" [fig. 4.3]. Ad ogni ciclo iterativo, ci si sposta nella cartella "data" del programma ESP, nella quale si trova il file "baseline_mod.csm". Quest'ultimo viene aperto e i cinque parametri vengono modificati cambiando la riga corrispondente con il valore di campionamento. I file step (o "stp") vengono generati nella cartella "bin" di ESP dopo che il terminale dedicato del programma compila il file CSM. Per fare in modo che ogni file step non venga sovrascritto, ad ogni iterazione si modifica anche la linea che specifica il nome del file step da creare. Infine, una volta modificate tutte le righe necessarie, viene chiuso il file CSM e avviato ESP senza interfaccia grafica tramite il comando: "serveESP -batch", specificando il file baseline_mod.csm da leggere.

Tutte le istruzioni appena descritte vengono eseguite dal codice python riportato in Appendice A - codice 1. E' importante notare che il codice va eseguito dal terminale di ESP e non dal terminale del sistema operativo.

1	DP,wing:aspect,wing:thickr,wing:xroot,wing:alphab,fuse:power				
2	1,9.031410524234968,0.18215051761436296,51.64441923808042,2.2071812113161045,3.388312810945145				
3	2,9.601576695869852,0.15796314268451261,56.464707382094254,1.8314660340283584,2.27770714941101				
4	3,10.811417350980914,0.19274696620688828,51.59783802196931,2.6188695945073066,3.076971044613331				
5	4,8.998955794147435,0.13337142846704697,54.12929422453629,1.0260655053585912,2.5290510606287016				
6	5,9.912497457709481,0.13187358537852711,54.55605589593861,2.9873884313934247,3.2100509947740417				
7	6,7.347288313820131,0.16268616635186117,51.87107508030652,2.9367383702436536,2.212241493981179				
8	7,8.483656201856828,0.17692073482210713,54.9465806608021,2.4877108464598434,2.94162349460283				
9	8,10.097061757676576,0.1274997276206636,51.31982785979421,2.844619565148296,2.4984912503090566				
10	9,8.412125143979829,0.1646777644762231,52.78670693364412,1.9916718015777894,2.772866588901336				
11	10,7.54206407313038,0.119165189736232,55.04499029015699,1.4848702997517635,2.1382718227788327				
12	11,9.335197414837431,0.14201730988719846,53.96169997695195,1.3501707070526674,3.250777827815038				
13	12,8.273376219694477,0.12123021348056552,55.87183342312,2.7507205298430684,2.869359976896842				
14	13,9.831490797398956,0.12563429191048273,51.727193054887834,1.0942008570524566,2.8831242592000095				
15	14,10.485283273034483,0.16764793346685442,55.09671416069141,1.138706030982852,2.0847520118987757				
16	15,8.135329300257148,0.19504586520820563,54.70171064984396,1.8682371720015518,3.964274431521014				
17	16,10.188466666458988,0.11731446035411168,53.33488713891426,2.4150477843853024,2.0372914198821577				
18	17,10.952234624315151,0.1809698427447543,56.1066530269204,1.6760376766167717,3.6713728528029943				
19	18,8.872211428156325,0.173340798571758,52.671669753947,1.3823723359861664,3.3014676437638757				
20	19,7.777629045736804,0.13644251433930268,51.98363538606464,1.6385626881768869,3.7998429896013084				

Figura 4.3: Valori dei parametri per i primi 19 DP

4.3 Preparazione della geometria e della mesh

Prima di generare la mesh sulla configurazione di base, è necessario creare il dominio di calcolo. Tramite il Workbench di Ansys viene importato il file "baseline_mod.stp" e avviato DesignModeler, uno dei software di modellazione presenti in Ansys. A differenza del condotto in cui si aveva un problema di fluidodinamica interna, nel caso del velivolo si affronta un problema di fluidodinamica esterna. I tre comandi principali che permettono di creare il dominio fluido sono i seguenti:

- **Enclosure:** comando disponibile tra gli strumenti principali che permette di generare un dominio computazionale attorno all'aereo, come si può vedere in figura 4.4.

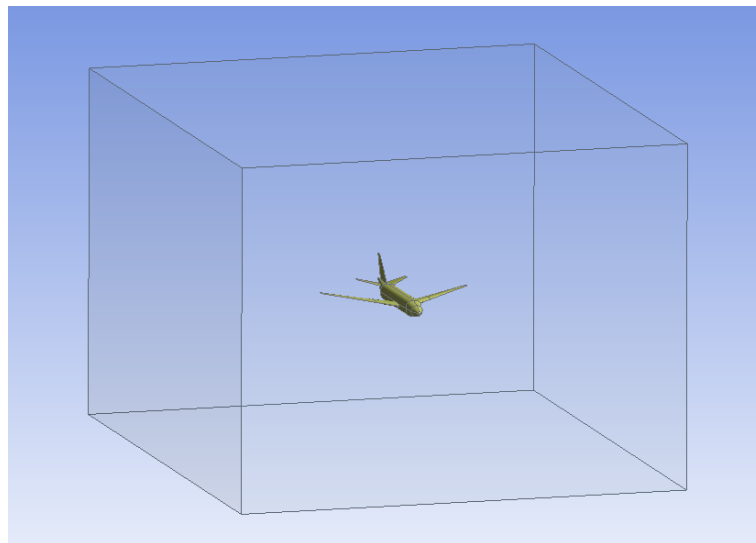


Figura 4.4: Enclosure

- **Symmetry:** tool che permette di dividere tutto il dominio a metà rispetto al piano XZ. Così facendo si cerca di risparmiare volume in modo da ridurre i tempi di simulazione.
- **Boolean - subtract:** questo strumento consente di rimuovere l'aereo solido dal dominio, lasciando solo il dominio fluido di interesse.

Il risultato finale è visibile di seguito [fig. 4.5]:

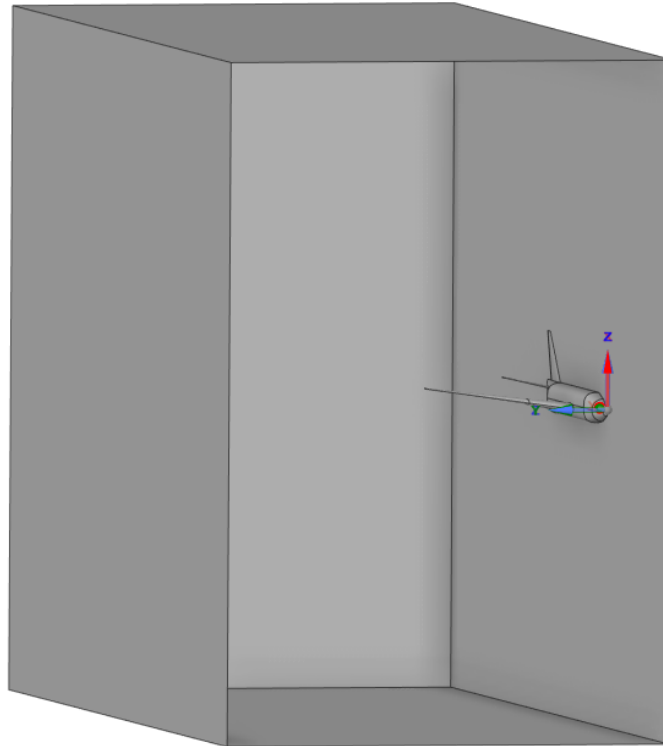


Figura 4.5: Dominio di calcolo

L'asse x identifica l'asse di rollio ed ha verso entrante nel dominio. L'asse z , o di imbardata, è positiva verso l'alto, mentre l'asse y è positiva nella direzione di rastremazione dell'ala destra. Il dominio è costituito da diverse zone:

- **Inlet**: superficie frontale al velivolo;
- **Outlet**: superficie posteriore;
- **Side**: superficie superiore, inferiore e laterale;
- **Symmetry**: piano di simmetria;
- **Plane**: superfici relative all'aeromobile;

La mesh è stata creata utilizzando Fluent Meshing, attraverso questi step:

- 1) la geometria viene importata tramite *watertight geometry*;
- 2) si genera la mesh di superficie;
- 3) viene definita la tipologia delle regioni fluide;
- 4) viene inserito uno strato limite tramite *inflation*;
- 5) si genera la mesh di volume

Dopo aver svolto uno studio di convergenza, si ottiene la mesh visibile in figura 4.6 con le seguenti proprietà:

Caratteristiche principali della griglia			
Numero di celle: 425052			
Numero di facce: 2199081			
Numero di nodi: 1479730			
Surface Mesh		Volume Mesh	
Minimum Size [mm]	0.18	Type	Polyhedra
Maximum Size [mm]	50	Max Cell Length [mm]	50
Growth Rate	1.2	Growth Rate	1.2
Size Functions	Curvature & Proximity		
Boundary Layer		Quality	
Number of Layers	8	Max. Skewness	0.8998
Growth Rate	1.1	Min. Orthogonal Quality	0.1002
First Height [mm]	0.1	Max. Aspect Ratio	43.9261
Grow on	Plane	y+	< 1

Tabella 4.3: Caratteristiche della mesh

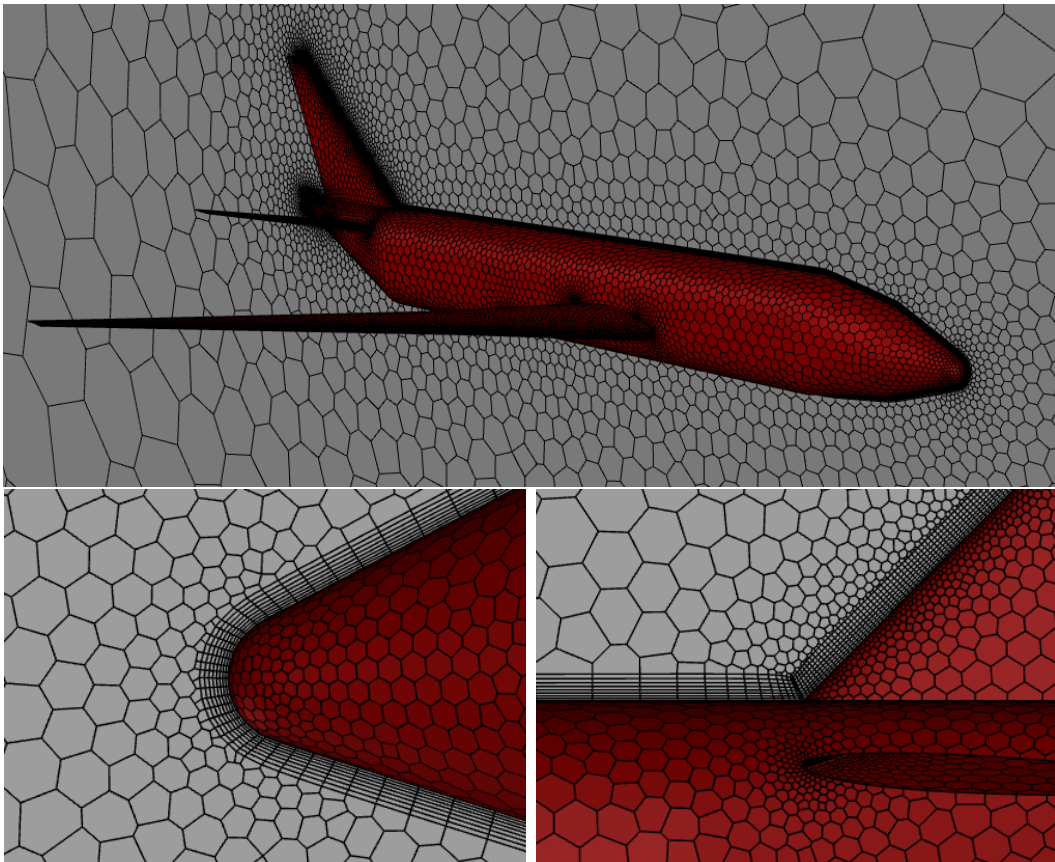


Figura 4.6: Mesh utilizzata

4.4 Simulazione baseline

Si vuole simulare l'aeromobile in una situazione di salita stazionaria. Per identificare una condizione di volo il più possibile realistica si è fatto uso dell'app di Flightradar24¹. Si è potuto osservare che i velivoli B-787-800 si portano alla quota di crociera con un rateo di salita che varia dai 1500 ft/min ai 5000 ft/min circa. Ipotizzando una velocità costante di 120 m/s e un rateo di salita di 2000 ft/min, si ricava un angolo di rampa di 4.8°. Si utilizza questo stesso valore anche per l'angolo di attacco α .

Si presuppone inoltre una quota istantanea di 7000 piedi. Le proprietà dell'aria relative a tale condizione di volo possono essere calcolate utilizzando il modello di

¹<https://www.flightradar24.com/>

atmosfera standard ISA76. Attraverso il sito di Digital Dutch², si ottiene:

Input	
Altitude	7000 ft
Temperature offset	0 K
Output	
Temperature	274.282 K
Pressure	78185.4 Pa
Density	0.993040 kg/m ³
Speed of sound	332.004 m/s
Dynamic viscosity	1.74202 e-5 Pa s

Tabella 4.4: Dati dell'aria a 7000 ft

La mesh viene importata in Fluent e si procede alla fase di set-up. Vengono impostate le seguenti opzioni principali:

- Simulazione stazionaria;
- Solutore density-based;
- Modello di turbolenza k-omega SST;
- Aria come gas ideale e legge di Sutherland per la viscosità;
- Condizioni al contorno:
 - **Inlet** [pressure-far-field]: velocità del flusso d'aria di 120 m/s (Mach 0.36) inclinato di $\alpha = 4.8^\circ$. Pressione e temperatura relative alla quota stabilita. L'angolo di traiettoria (o di rampa) non viene considerato per la direzione del flusso;

²<https://www.digitaldutch.com/>

- **Outlet** [pressure-outlet]: pressione e temperatura relative alla quota stabilita, così come all’ingresso
 - **Side** [pressure-far-field]: stesse condizioni dell’*Inlet*;
 - **Symmetry** [symmetry]: piano di simmetria;
 - **Plane** [wall];
- Parametri di monitoraggio: si osserva l’andamento del coefficiente di resistenza e la portata d’aria in uscita per verificare la convergenza della soluzione;
 - Metodi di soluzione:

Solution Methods	
Formulation	Implicit
Flux Type	Roe-FDS
Spatial Discretization	
Gradient	Least Squares Cell Based
Flow	First/Second Order Upwind
Turbulent Kinetic Energy	First/Second Order Upwind
Specific Dissipation Rate	First/Second Order Upwind
Other Settings	
Pseudo Time Method	Off
Warped-Face Gradient Correction	On
High Order Term Relaxation	Off
Other Settings	
Courant Number	1 to 50

Tabella 4.5: Metodi di soluzione

Il numero di Courant (CFL) rappresenta un fattore molto importante per la stabilità numerica delle applicazioni CFD. Viene definito come:

$$CFL = \frac{c\Delta t}{\Delta x} \quad (4.1)$$

dove c rappresenta la velocità del flusso, Δx l'intervallo spaziale e Δt l'intervallo temporale. In generale, l'uso di intervalli temporali più ampi porta ad una convergenza più rapida, quindi è vantaggioso impostare un numero di Courant elevato. Il CFL può raggiungere anche un valore prossimo a 100 per metodi impliciti. Tuttavia, è buona norma aumentare il numero di Courant progressivamente mentre la soluzione avanza, evitando così problemi di instabilità e divergenza.

- Inizializzazione ibrida e inizializzazione full-multigrid (opzionale);

COURANT NUMBER	ITERATIONS
First/Second Order Upwind	
1	100
2	50
3	50
4	50
5	50
7	50
10	50
12	50
15	50
20	50
25	50
30	50
40	50
50	100

Tabella 4.6: Strategia di convergenza

Per questo problema la simulazione è stata eseguita in modo diverso. La complessità della geometria e della griglia ha richiesto una diversa strategia di convergenza [tab. 4.6]. Innanzitutto si è utilizzato un grado di precisione inferiore (First Order Upwind). Variando il numero di Courant da 1 a 50, dopo alcune iterazioni, si è ottenuta una rapida convergenza. Successivamente, si è aumentato il grado di accuratezza al secondo ordine (Second Order Upwind) e, con la medesima procedura, abbiamo ottenuto il risultato desiderato con residui dell'ordine di $1e-4$.

Tutti questi risultati sono visibili dal grafico dei residui:

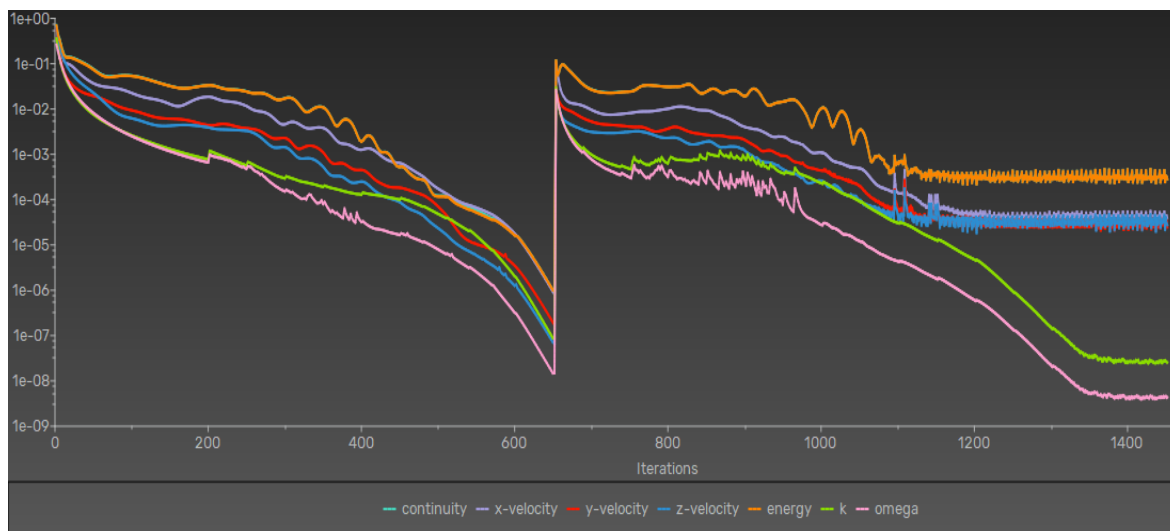


Figura 4.7: Residui

Infine, i parametri di monitoraggio forniscono un'ulteriore verifica.

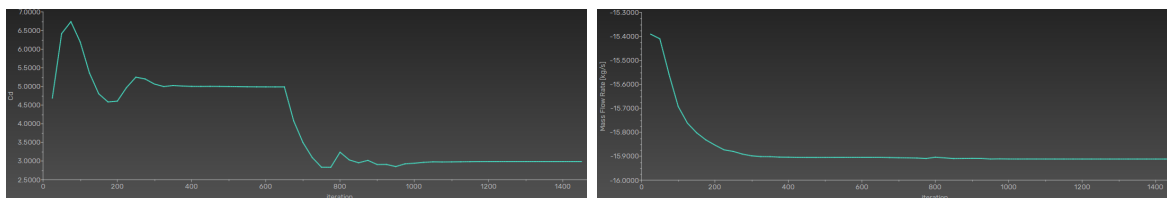
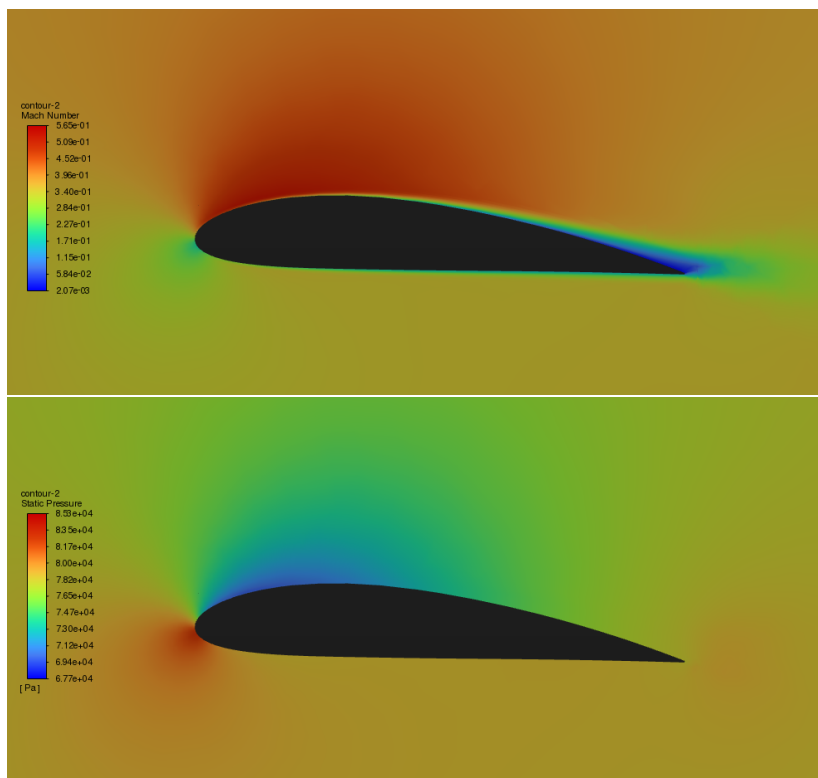


Figura 4.8: Parametri di controllo: C_d e portata all'Outlet

In figura 4.9a e 4.9b si possono osservare i grafici di contorno per la velocità, espressa in numero di Mach per alcune zone caratteristiche. Il primo grafico mostra la velocità sul piano di simmetria ed è possibile notare come la velocità diminuisca in prossimità delle pareti, sulla parte anteriore e posteriore. Il secondo grafico conferma il corretto funzionamento del profilo alare, marcando la zona superiore di depressione e la zona inferiore di sovrappressione. Con un angolo di attacco di 4.8° ci si attendeva tale situazione. Sulla parte terminale del profilo si può notare l'inizio della separazione dello strato limite, fenomeno che sarebbe molto più marcato a velocità maggiori a parità di angolo di attacco.



(a) Contour della velocità del velivolo nel piano di simmetria



(b) Contour della pressione e del numero di Mach per un profilo alare vicino alla radice

Figura 4.9: Linee di contorno - zone principali

Calcolando le forze in direzione del flusso alla quota di 7000 piedi e considerando come superficie di riferimento

$$S = \frac{4240 \text{ mm}^2}{2} = 2120 \text{ mm}^2$$

si ottengono i seguenti risultati:

Direction Vector	(0.9965 0 0.0837)	Direction Vector	(-0.0837 0 0.9965)
Drag force	1.0610 [N]	Lift	10.2919 [N]
Drag coefficient	0.0706	Lift coefficient	0.6855

Per una valutazione rapida dei risultati, consideriamo il caso reale corrispondente. Si suppone una tratta breve ed una massa di 180 000 kg.

Una delle equazioni fondamentali della manovra di salita stazionaria prevede:

$$L = W \cos(\gamma) = m g \cos(\gamma)$$

in cui W rappresenta il peso del velivolo, γ l'angolo di rampa e L la forza di portanza.

Esprimendo la portanza come:

$$L = \frac{1}{2} \rho V^2 S c_L$$

si può ricavare un'approssimazione sul coefficiente di portanza.

$$c_L = \frac{2 W \cos(\gamma)}{\rho V^2 S} \simeq 0.65$$

Il valore di 0.6855 trovato dalla simulazione può considerarsi ragionevole, così come il coefficiente di resistenza. Si sottolinea che i risultati aerodinamici ricavati da questo modello semplificato sono ragionevoli e coerenti dal punto di vista fisico ma, nonostante ciò, non possono essere considerati precisi ed affidabili.

L'efficienza aerodinamica, definita come il rapporto tra portanza (L) e resistenza (D), risulta essere pari a:

$$E = \frac{L}{D} = \frac{c_L}{c_D} = 9.7306$$

4.5 Workflow completo - velivolo

La simulazione di tutti i DP avviene in egual modo con tutto ciò già descritto in sezione 3.6 del capitolo 3. Il codice che gestisce tutta la procedura è riportato in appendice A - Codice 2.

La differenza sostanziale di questo codice rispetto a quello utilizzato per il condotto è osservabile nelle righe di codice successive alla creazione della cartella di lavoro e alla copia dei file necessari. La scelta di dividere per simmetria il dominio di calcolo in modo da risparmiare numero di celle e tempo di soluzione, comporta la necessità di suddividere a metà sia la baseline che il modello esaminato. Ad ogni iterazione, perciò, sia la baseline che la configurazione modificata vengono bipartiti attraverso comandi della libreria di CadQuery. In particolare, si fa uso della distanza tra i due profili alari alle estremità, ovvero l'apertura alare totale (b), la quale è collegata al rapporto d'aspetto (A) dell'ala dalla seguente relazione:

$$A = \frac{b^2}{S}$$

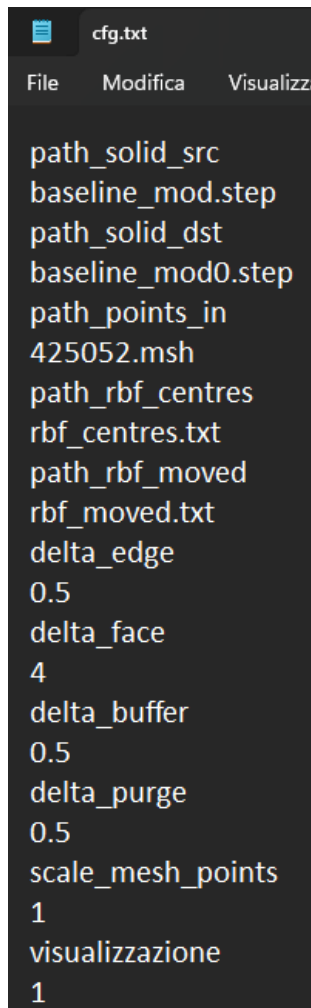
Si ricorre al piano parallelo al profilo alare di estremità traslato di uno spostamento in modulo pari a $b/2$ in direzione $-y$. Per la configurazione di base questa distanza è fissata in quanto $A = 9$ e $S = 4240 \text{ mm}^2$ e viene indicata con $x1$.

$$x1 = \frac{\sqrt{9 \cdot 4240}}{2}$$

Con la stessa formula si ricava anche la distanza $x2$ relativa al velivolo modificato. Siccome uno dei parametri di ottimizzazione è proprio il rapporto d'aspetto dell'ala, è necessario leggere ad ogni ciclo il valore di A corrispondente dal file CSV "points_LHS_plane.csv". Con il comando "df.iloc[i,1]" viene estratto il valore richiesto dal data-frame. Dopo la divisione vengono esportati i file con lo stesso nominativo ma con

estensione ".step" per non sovrascrivere i precedenti.

Il numero di celle della mesh è stato utilizzato per nominare la mesh, il caso di studio ed i risultati in Fluent per la baseline (425052.msh, 425052.cas.h5 e 425052.dat.h5). In figura 4.10 e 4.11 si riportano rispettivamente il file di input per l'eseguibile *roc.exe* e il file journal da leggere in ambiente Fluent.



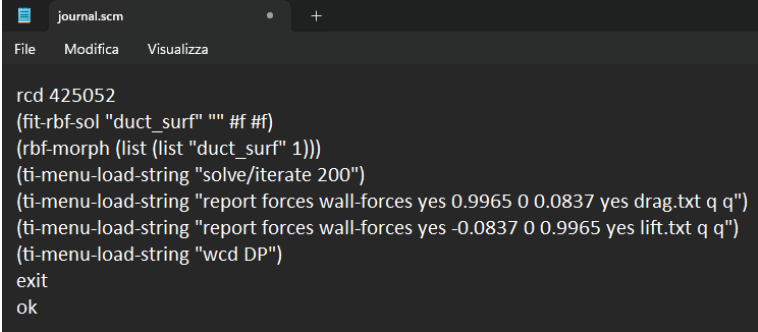
```
cfg.txt
File Modifica Visualizz

path_solid_src
baseline_mod.step
path_solid_dst
baseline_mod0.step
path_points_in
425052.msh
path_rbf_centres
rbf_centres.txt
path_rbf_moved
rbf_moved.txt
delta_edge
0.5
delta_face
4
delta_buffer
0.5
delta_purge
0.5
scale_mesh_points
1
visualizzazione
1
```

Figura 4.10: cfg.txt - velivolo

Per la generazione della nuvola di punti del velivolo, la densità di punti sui bordi e sulle facce è stata aumentata rispetto al caso di studio del condotto. La geometria, essendo più complessa così come la mesh, richiede una maggiore risoluzione, soprattutto

nei bordi. La configurazione di questo file cfg ha avuto successo per la maggior parte dei design points. Per altri DP più critici si è dovuto aumentare il *delta_face* in modo da aumentare la flessibilità della mesh.

A screenshot of a terminal window with a dark background. The title bar shows 'journal.scf' and a plus sign. The menu bar contains 'File', 'Modifica', and 'Visualizza'. The terminal text is as follows:

```
rcd 425052
(fit-rbf-sol "duct_surf" "" #f #f)
(rbf-morph (list (list "duct_surf" 1)))
(ti-menu-load-string "solve/iterate 200")
(ti-menu-load-string "report forces wall-forces yes 0.9965 0 0.0837 yes drag.txt q q")
(ti-menu-load-string "report forces wall-forces yes -0.0837 0 0.9965 yes lift.txt q q")
(ti-menu-load-string "wcd DP")
exit
ok
```

Figura 4.11: journal.scf - velivolo

Il file journal, a differenza del condotto, esegue 200 iterazioni dopo aver deformato la mesh. Infine vengono salvate le forze in direzione parallela al flusso e in direzione ortogonale ad esso.

4.6 Modello di esempio

Per un'analisi più approfondita si è scelto di utilizzare i risultati ottenuti per il design point numero 2 che corrisponde al DP1. Il velivolo modificato presenta soprattutto un allungamento alare maggiore, uno spostamento dell'ala verso la coda ed una fusoliera con forma molto più circolare e cilindrica. Queste caratteristiche sono visibili nelle figure sottostanti.

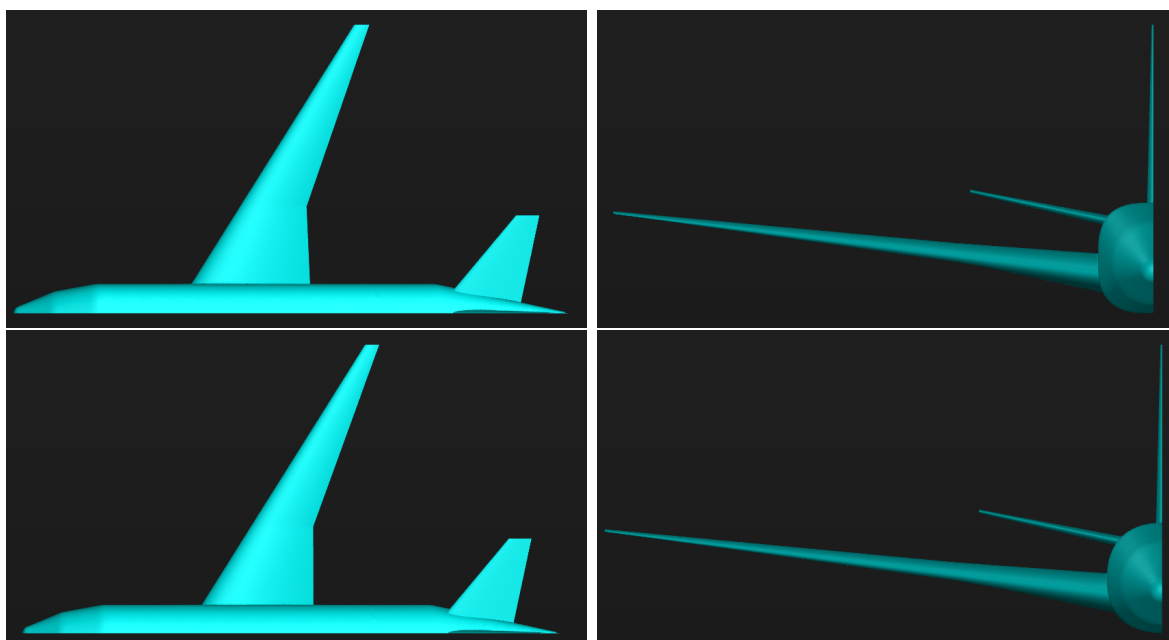


Figura 4.12: Geometrie a confronto: in alto la baseline, in basso il DP1

Le differenze sono ancor più visibili se si visualizzano i punti sorgente (in rosso) e i punti traslati (in verde), figura 4.13. Per quanto riguarda la mesh, la deformazione è contenuta e non presenta particolari problematiche come si può osservare nelle immagini 4.14. La prima immagine mette a confronto la griglia originale con quella ottenuta dopo il processo di morphing. La qualità per questo design point è totalmente preservata: la minima qualità ortogonale rimane pari a 0.1, mentre il rapporto d'aspetto massimo risulta essere 44.307.

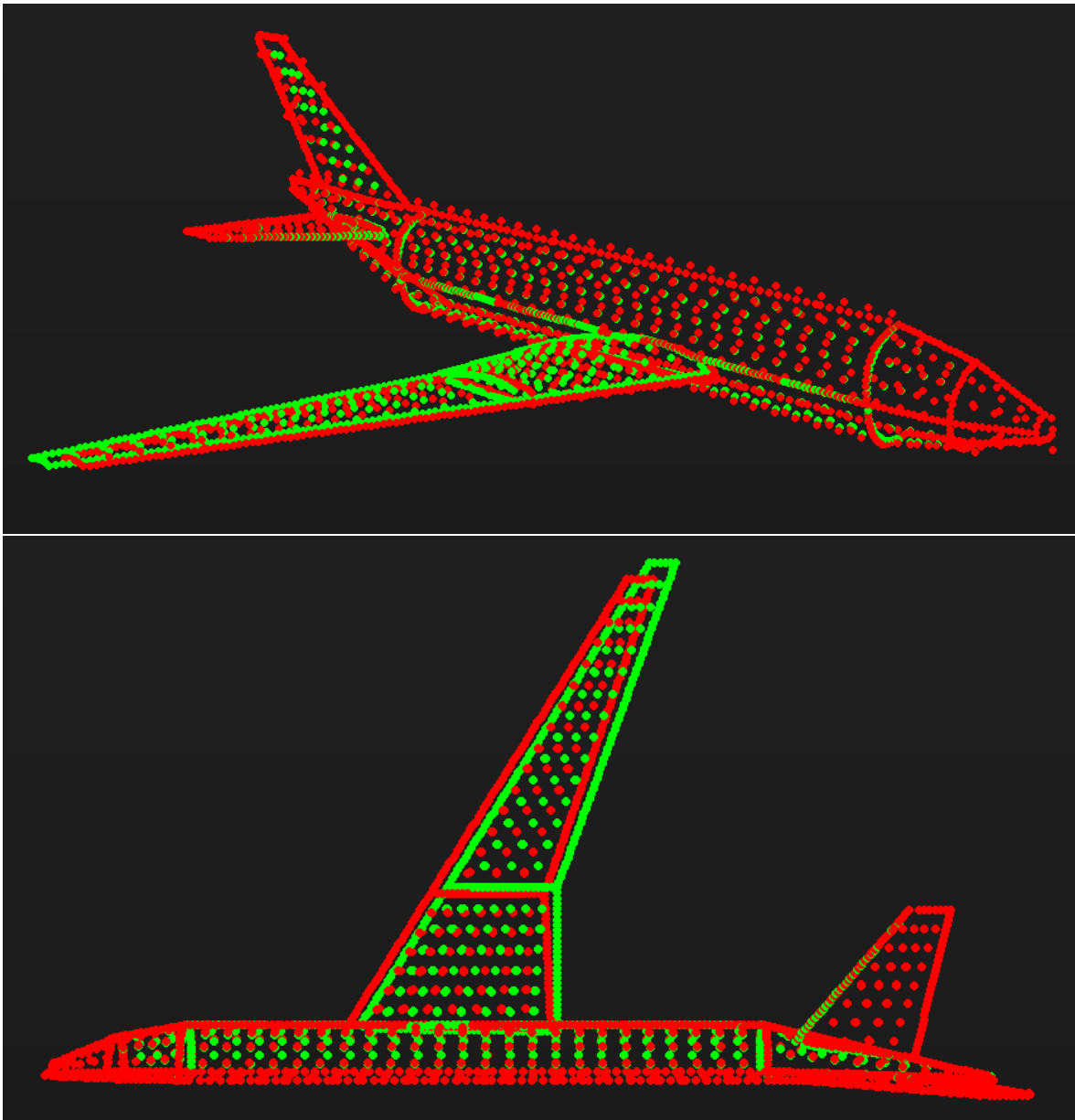
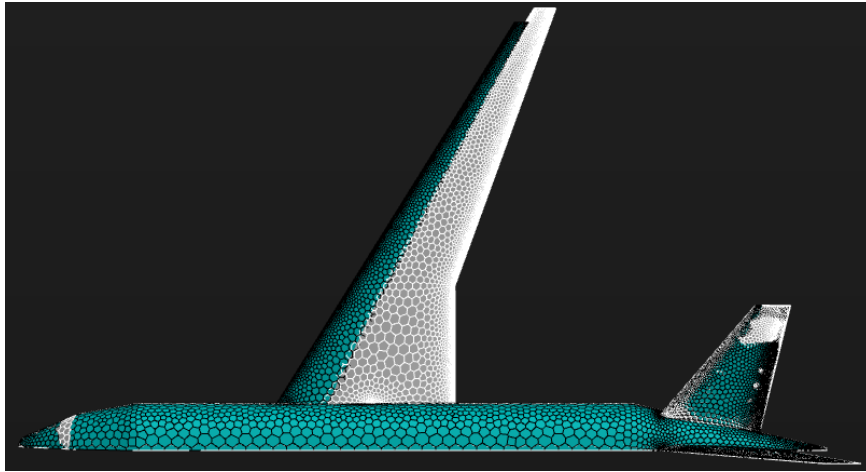
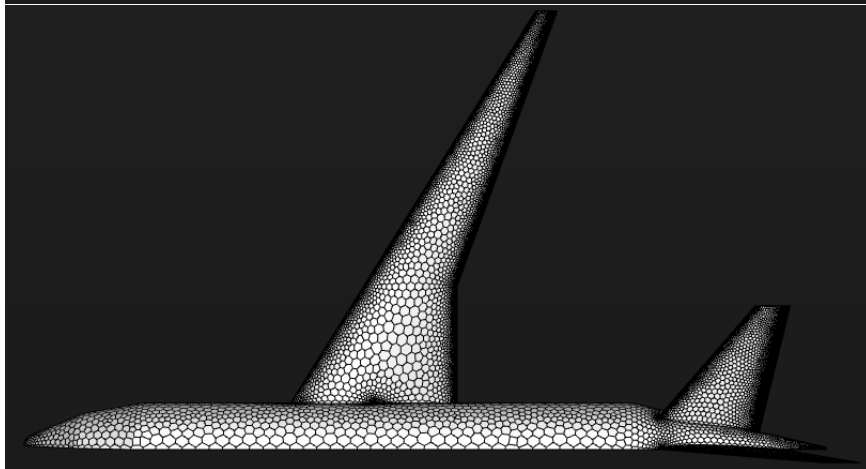
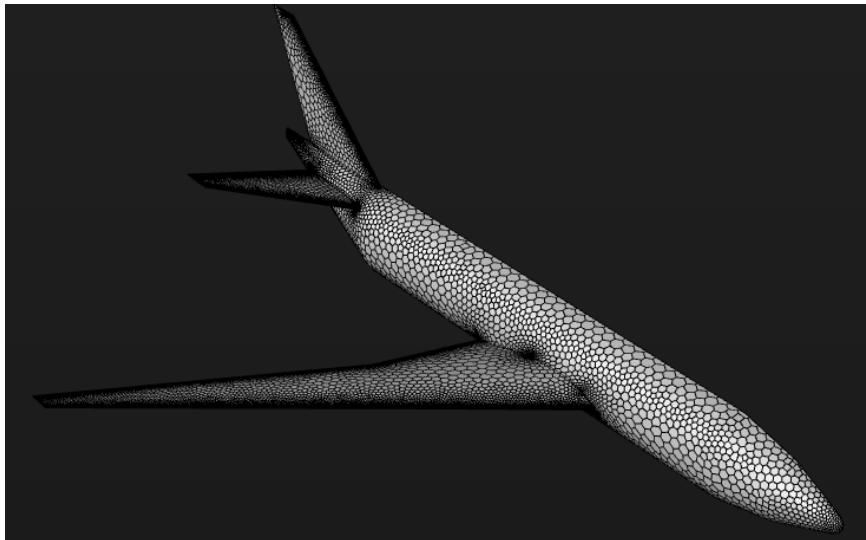


Figura 4.13: punti sorgente e punti finali



(a) Confronto tra mesh baseline (azzurra) e mesh finale (in bianco)



(b) mesh dopo il morphing

Figura 4.14: griglie a confronto

I risultati delle forze e dei coefficienti sono leggermente inferiori per questa specifica combinazione.

Drag force	1.0421 [N]	Lift	10.2241 [N]
Drag coefficient	0.0693	Lift coefficient	0.6804

Se invece si considera un modello con alcuni parametri prossimi al valore limite inferiore o superiore si avrà una riduzione maggiore della qualità. Se ad esempio si prende in esame il design point numero 50, la minima qualità ortogonale finale è di 0.01; tale valore è accettabile per questa simulazione di test, ma potrebbe essere troppo basso se si vuole condurre una simulazione ad alta fedeltà. Per evitare questa situazione la mesh dovrebbe essere molto più fitta e precisa, soprattutto nelle zone in cui la geometria è più complessa. In questo modo, anche per deformazioni ampie della geometria, la mesh riesce meglio ad adattarsi ai cambiamenti.

4.7 Ottimizzazione

Per eseguire l'ottimizzazione di forma è necessario creare un file CSV aggiornato con i risultati, parallelamente a quanto svolto con il condotto. Il codice 3 in appendice A preleva i dati delle forze dai file di output per ogni DP, ne calcola l'efficienza aerodinamica e la riporta nel data-frame. I dati finali vengono poi esportati in formato CSV con la dicitura "results.csv". Il parametro P6 rappresenta l'efficienza aerodinamica.

1	Name,P1,P2,P3,P4,P5,P6								
2	1,9.031410524234968,0.1821505176143629,51.64441923808042,2.2071812113161045,3.388312810945145,9.436037479001923								
3	2,9.601576695869852,0.1579631426845126,56.464707382094254,1.8314660340283584,2.27770714941101,9.811005374677698								
4	3,10.811417350980914,0.1927469662068882,51.59783802196931,2.6188695945073066,3.076971044613331,9.264355150766866								
5	4,8.998955794147435,0.1333714284670469,54.12929422453629,1.0260655053585912,2.5290510606287016,9.966213816284553								
6	5,9.91249745770948,0.1318735853785271,54.55605589593861,2.9873884313934247,3.2100509947740417,9.680846184710846								
7	6,7.347288313820131,0.1626861663518611,51.87107508030652,2.9367383702436536,2.212241493981179,9.463543917587389								
8	7,8.483656201856828,0.1769207348221071,54.9465806608021,2.4877108464598434,2.94162349460283,9.479129669974142								
9	8,10.097061757676576,0.1274997276206636,51.31982785979421,2.844619565148296,2.4984912503090566,9.857538043043911								
10	9,8.412125143979829,0.164677644762231,52.78670693364412,1.9916718015777892,2.772866588901336,9.634349840263196								

Figura 4.15: results.csv

Table of Outline A10: Design Points of Design of Experiments							
	A	B	C	D	E	F	G
1	Name	P1 - wing:aspect	P2 - wing:thidr	P3 - wing:xroot	P4 - wing:alphab	P5 - fuse:power	P6 - Efficiency
2	1	9,0314	0,18215	51,644	2,2072	3,3883	9,436
3	2	9,6016	0,15796	56,465	1,8315	2,2777	9,811
4	3	10,811	0,19275	51,598	2,6189	3,077	9,2644
5	4	8,999	0,13337	54,129	1,0261	2,5291	9,9662
6	5	9,9125	0,13187	54,556	2,9874	3,2101	9,6808
7	6	7,3473	0,16269	51,871	2,9367	2,2122	9,4635
8	7	8,4837	0,17692	54,947	2,4877	2,9416	9,4791
9	8	10,097	0,1275	51,32	2,8446	2,4985	9,8575
10	9	8,4121	0,16468	52,787	1,9917	2,7729	9,6343

Figura 4.16: risultati importati nel DOE del blocco "Response Surface Optimization" del Workbench

Successivamente, utilizzando l'ambiente del Workbench, tutti i DP vengono importati per lo studio del DOE. Nella figura 4.17 è possibile osservare la distribuzione delle ampiezze dei sei parametri per tutti i design points: se si prende in considerazione un determinato parametro (linee verticali), ogni DP ne assume un valore diverso, creando una spaziatura quasi uniforme.

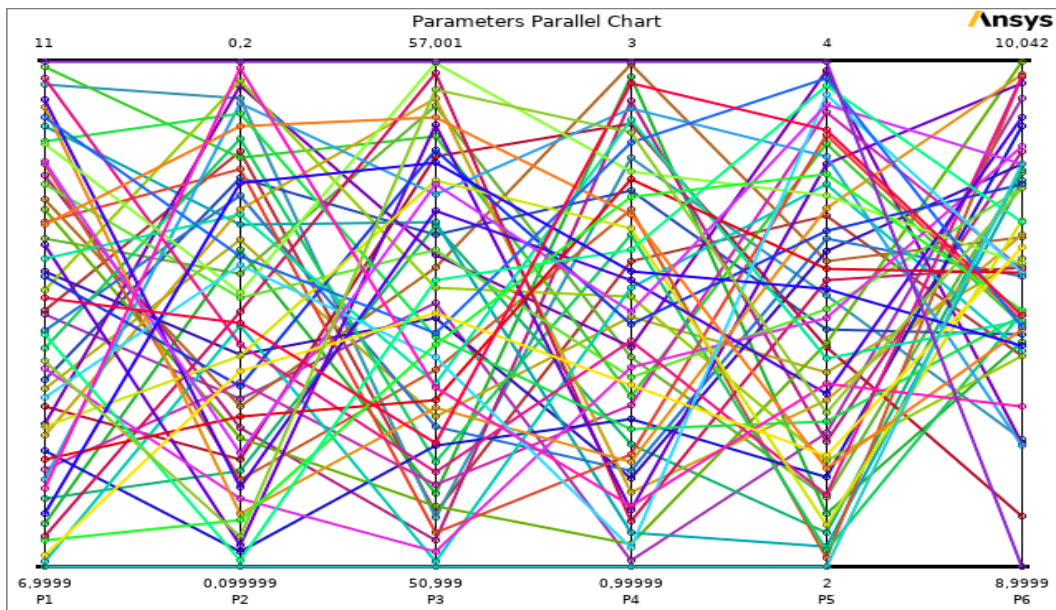


Figura 4.17: distribuzione delle ampiezze

Attraverso la ricerca del minimo e del massimo per il parametro di output, si ottengono varie superfici di risposta. In figura 4.20 ne sono state riportate alcune di esempio. L'obiettivo dell'ottimizzazione in questione è massimizzare l'efficienza aerodinamica:

Table of Schematic A4: Optimization									
	A	B	C	D	E	F	G	H	I
1	Name	Parameter	Objective			Constraint			
2			Type	Target	Tolerance	Type	Lower Bound	Upper Bound	Tolerance
3	Maximize P6	P6 - Efficiency	Maximize	0		No Constraint			

Figura 4.18: Obiettivo: massimizzare l'efficienza

Table of Schematic A4: Optimization , Candidate Points									
	A	B	C	D	E	F	G	H	I
1	Reference	Name	P1 - wing:aspect	P2 - wing:thickr	P3 - wing:xroot	P4 - wing:alphab	P5 - fuse:power	P6 - Efficiency	
2								Parameter Value	Variation from Reference
3	●	Candidate Point 1	10,556	0,1105	52,851	1,0326	2,0022	★ ★ ★ 10,207	0,00%
4	○	Candidate Point 2	10,63	0,10994	52,905	1,0128	2,0016	★ ★ ★ 10,207	0,00%
5	○	Candidate Point 3	10,516	0,11058	52,866	1,0166	2,0038	★ ★ ★ 10,207	0,00%

Figura 4.19: punti candidati - velivolo

La geometria ottimale secondo i punti candidati possiede un allungamento alare poco inferiore a quello massimo imposto. Di fatti, questo parametro ha effetto sul

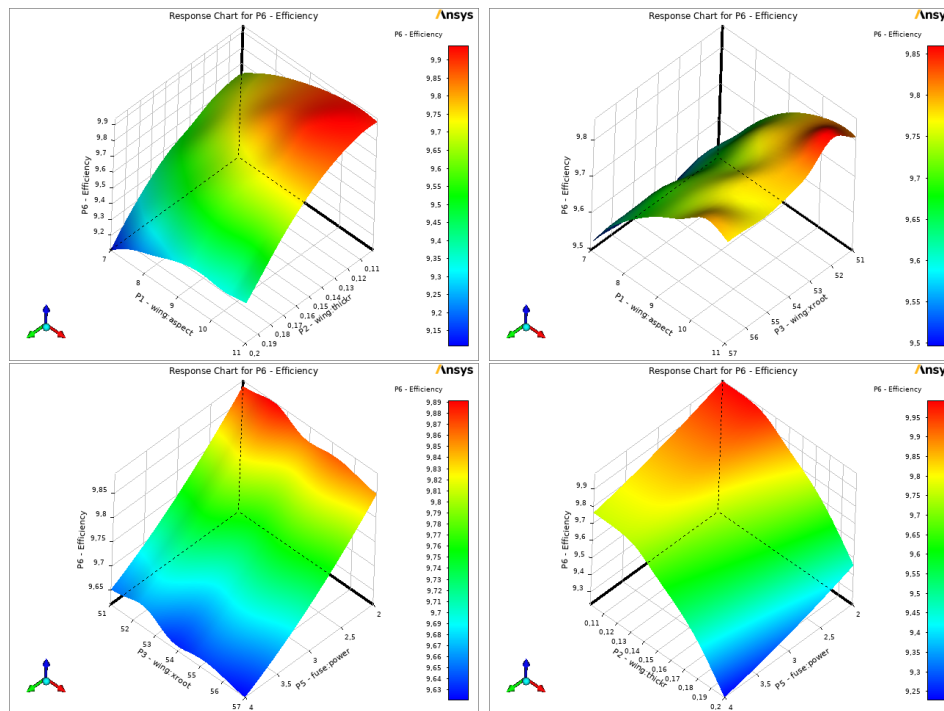


Figura 4.20: Alcune superfici di risposta 3D

coefficiente di resistenza indotta (c_{Di}). La resistenza totale del velivolo si può suddividere principalmente in un contributo di resistenza viscosa (c_{D0}) e in uno di resistenza indotta dalla portanza a causa della lunghezza finita dell'ala. Un ulteriore contributo (resistenza d'onda) si manifesta ad alte velocità di volo quando, in almeno un punto della superficie dell'aeromobile, la velocità dell'aria è maggiore della velocità del suono locale. Siccome ci si trova a basse velocità, si trascurano gli effetti della comprimibilità. Una formula che permette il calcolo del coefficiente di resistenza di un velivolo con ragionevole accuratezza è sicuramente la polare di Prandtl:

$$c_D = c_{D0} + c_{Di} = c_{D0} + \frac{c_L^2}{\pi A e} \quad (4.2)$$

dove e prende il nome di fattore di Oswald. E' un parametro che tiene conto della distribuzione di portanza sull'ala. Per altri dettagli fare riferimento a [7]. La formula 4.2 evidenzia come un allungamento alare maggiore diminuisca gli effetti di resistenza

indotta. Tuttavia, ali con rapporto d'aspetto elevato non sono semplici dal punto di vista strutturale. Inoltre, aumentando A , il velivolo diventa più sensibile alle raffiche. Infine, si può affermare che un allungamento alare maggiore è un risultato che ci si attendeva. In modo intuitivo, ci si aspettava anche una fusoliera più cilindrica e aerodinamica; questo è il motivo per cui la potenza della super-ellisse proposta è pari a 2. La posizione longitudinale dell'ala non si discosta eccessivamente dal valore originale. Al contrario, sia l'angolo di incidenza del profilo intermedio dell'ala sia lo spessore del profilo alla radice assumono valori prossimi a quelli dei limiti inferiori. In conclusione, l'influenza dei parametri nei confronti dell'efficienza aerodinamica è visibile nell'immagine in basso.

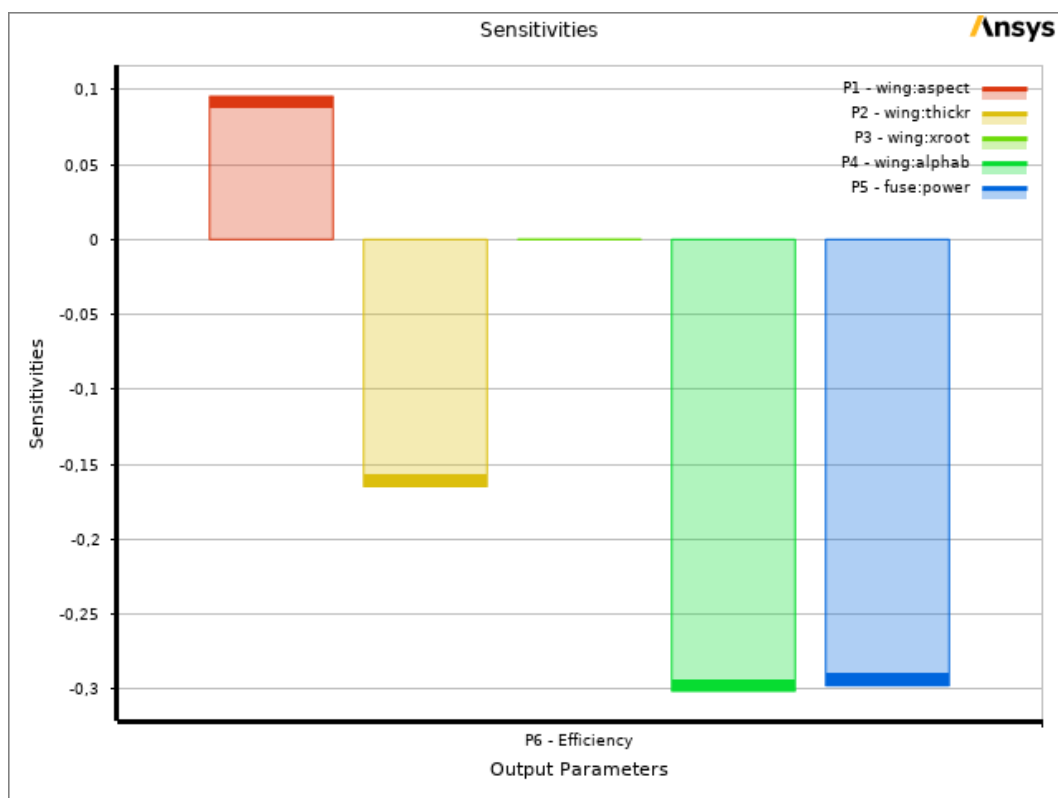


Figura 4.21: Sensibilità locale

Una volta terminata la procedura di ottimizzazione, viene esaminato il primo punto candidato. La geometria ottimale viene generata avviando il terminale di ESP e leggen-

do il file CSM con gli opportuni parametri aggiornati. In figura 4.22 viene confrontata la baseline con il modello ottimale.

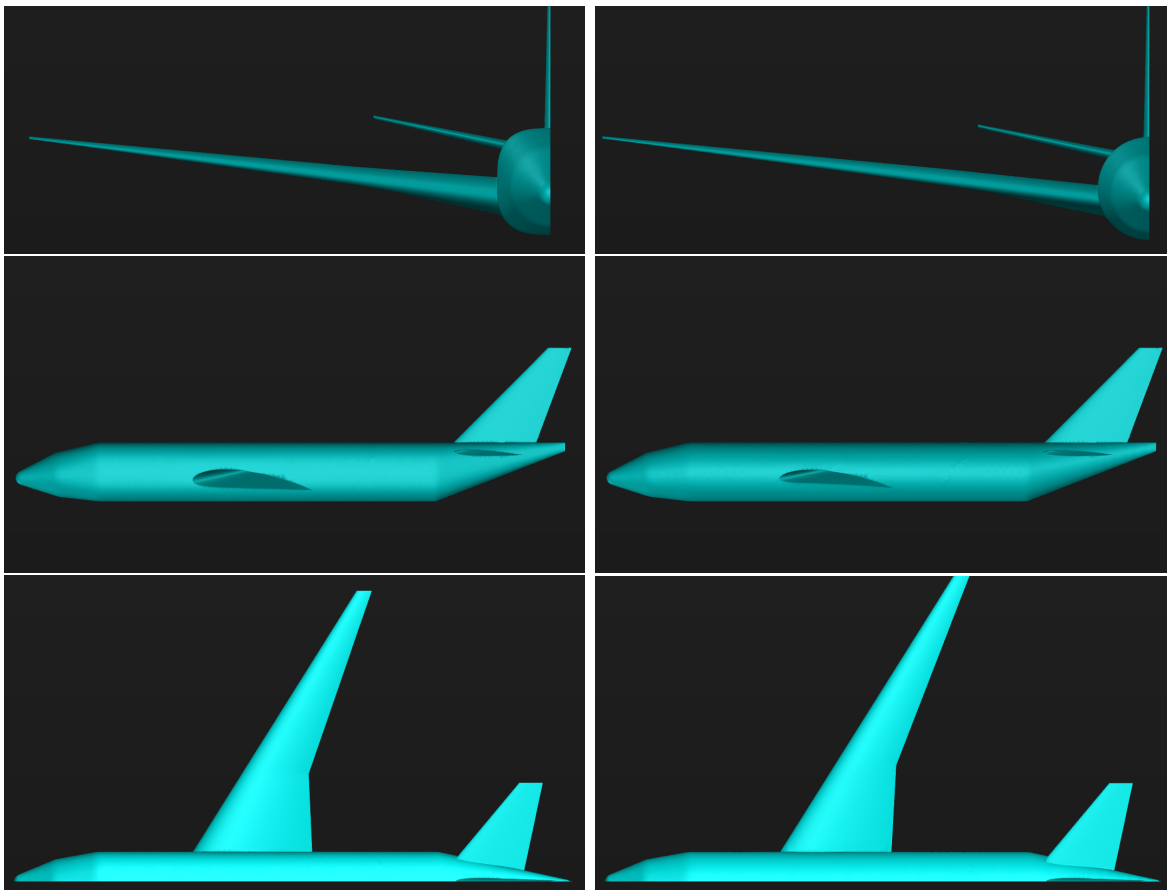
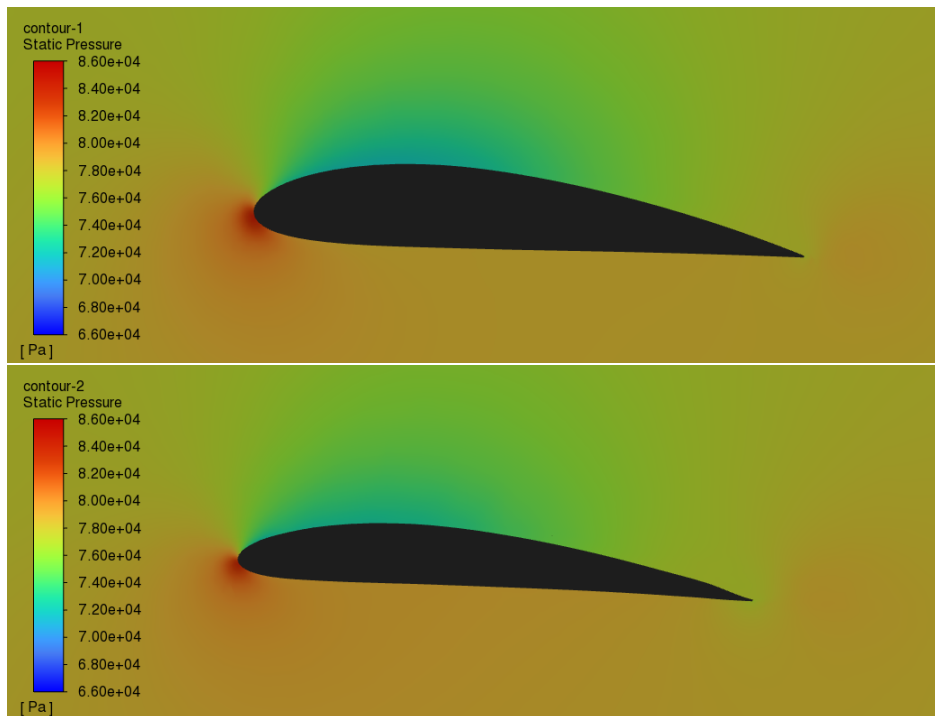
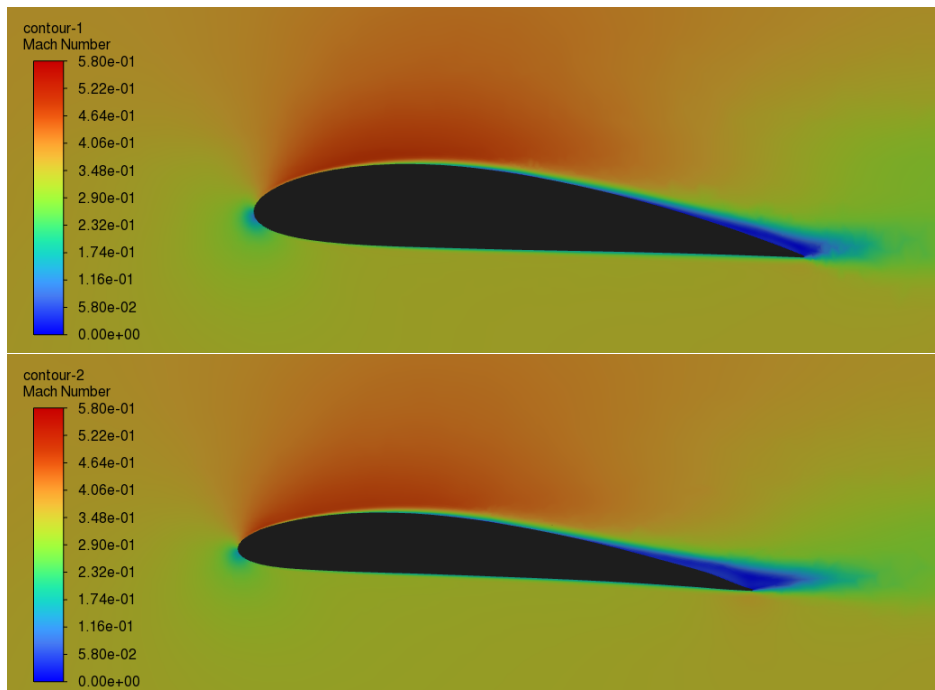


Figura 4.22: a sinistra la geometria baseline, a destra la geometria ottimizzata

I campi di pressione e velocità, espressa in numero di Mach, sono visibili nelle figure 4.23 e 4.24. Il profilo alare vicino all'incastro con la fusoliera subisce una contenuta riduzione della differenza di pressione tra dorso e ventre, con conseguente leggera diminuzione del coefficiente di portanza totale. La separazione in zona di scia è ristretta in entrambi i casi e non comporta particolari cambiamenti.



(a) campo di pressione: baseline in alto, ottimo in basso



(b) campo di velocità (Mach number): baseline in alto, ottimo in basso

Figura 4.23: campo di pressione e velocità per un profilo alare prossimo alla radice dell'ala

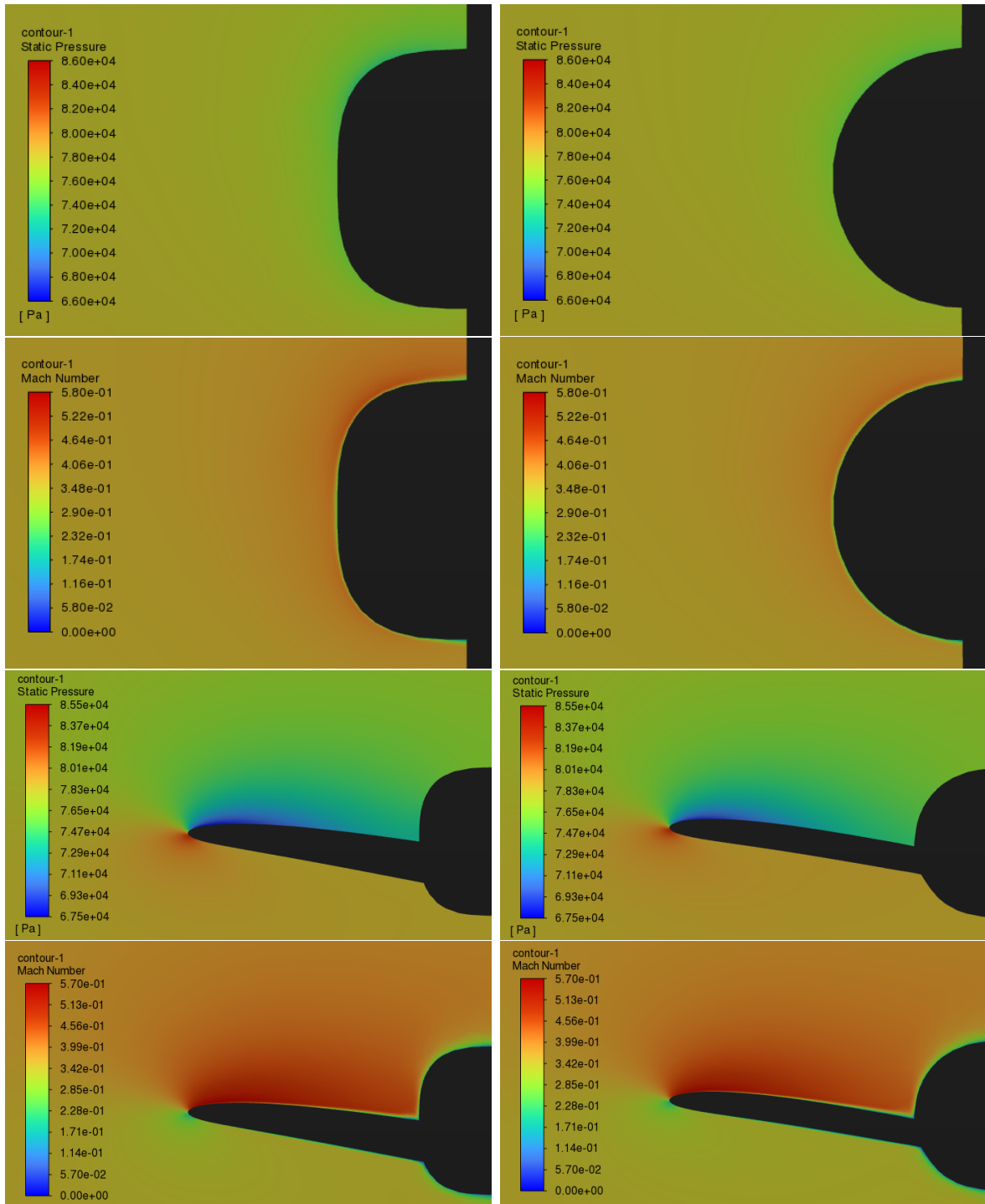


Figura 4.24: campo di pressione e velocità per la fusoliera: baseline a sinistra, ottimo a destra

Il cambio di forma della fusoliera consente di ridurre la forza di resistenza; la forma cilindrica risulta essere più affusolata e aerodinamica. Per il modello ottimale si ottengono questi risultati:

Drag force	0.9821 [N]	Lift	10.0935 [N]
Drag coefficient	0.0653	Lift coefficient	0.6717

Calcolando l'efficienza aerodinamica si ha:

$$E = \frac{L}{D} = \frac{10.0935}{0.9821} = 10.2774$$

Il valore atteso del parametro P6 dall'ottimizzazione numerica è visibile in figura 4.19 e vale 10.2070. Si può dunque concludere che il risultato ottenuto dalla simulazione CFD è in linea con quello previsto dal processo di ottimizzazione basato sulle superfici di risposta. I valori ottimizzati dei parametri scelti aumentano l'efficienza aerodinamica del 5.1% rispetto ai valori assegnati di default.

Capitolo 5

Conclusioni

Il lavoro ed i risultati illustrati in questa trattazione sono frutto di un processo di unificazione di molteplici strumenti software e funzionalità. I due casi di studio analizzati hanno permesso la creazione di un flusso di lavoro veloce ed automatizzato. I software CAD programmati per lavorare da riga di comando consentono di generare cicli iterativi in grado di formulare geometrie differenti in tempi molto rapidi. Il lavoro di impostazione e definizione dei parametri può comunque richiedere tempistiche considerevoli se si considera anche la mancata disponibilità delle competenze necessarie. Tuttavia, con un progetto parametrico si possono generare forme complesse che sarebbero difficili da ottenere con i metodi tradizionali. CadQuery ed Engineering Sketch Pad sono solo alcuni dei software di questo tipo presenti nel mercato. Un'altra particolarità di questi prodotti risiede nella possibilità di importare oggetti già progettati da altri utenti da una libreria virtuale.

RBF-Morph come strumento integrato in Fluent ha permesso di svolgere un'ottimizzazione di forma con una sola griglia di calcolo iniziale utilizzando il mesh morphing. Questo programma, così come Ansys Fluent, è eseguibile da riga di comando senza interfaccia grafica. Attraverso il linguaggio di programmazione Python è stato possibile connettere tutti questi software in modo dinamico e funzionale. Il condotto di test è

stato utile per capire il procedimento che sta alla base di questo studio. Per il velivolo, invece, è stato necessario apportare alcuni accorgimenti nei codici di sviluppo e di esecuzione.

L'eseguibile "roc.exe" è uno strumento ancora in fase di sperimentazione. Con questi due casi di studio si sono compresi i possibili miglioramenti e le criticità ancora presenti: lo strumento non è in grado di confrontare geometrie con numero di bordi o divisioni di essi differenti, quindi il numero di bordi e di superfici deve essere necessariamente il medesimo tra la baseline e il modello modificato. Per alcuni design points sono state riscontrate delle problematiche in fase di deformazione della mesh. In particolare, con alcune geometrie, si sono generate celle con volume negativo, le quali portano inevitabilmente a divergenza della soluzione. Queste casistiche sono state risolte modificando il numero dei punti sulle facce e sui bordi in base ad uno studio grafico della zona critica. Si è giunti alla conclusione che vi è la necessità di implementare un dimensionamento automatico iniziale in base alle dimensioni delle celle della mesh stessa.

Per quanto riguarda le tempistiche, considerando che la mesh utilizzata per il velivolo semplificato ha un numero di celle vicine al mezzo milione, la simulazione per la configurazione di base ha richiesto circa un'ora. Nel caso in cui si intende valutare le prestazioni del velivolo a velocità più elevate dove gli effetti della comprimibilità aumentano, il tempo dell'analisi aumenta. Per le analisi descritte in questa tesi, sono stati utilizzati 4 processori (core), sia per la fase di mesh, sia per l'analisi CFD. Circa 7 ore è stato il tempo necessario per il confronto delle geometrie, il morphing e le simulazioni di tutti i campioni. Alla fine del processo tutti i dati e le cartelle create hanno occupato circa 16 GB di spazio sul disco. Se alcuni dati copiati all'interno delle cartelle dei DPs non sono indispensabili, è opportuno eliminarli tramite python. Simulazioni più pesanti richiederebbero notevole spazio di archiviazione.

In conclusione si può affermare che, per svolgere un'ottimizzazione di forma in tempi

rapidi al fine di migliorare prestazioni e zone critiche di un oggetto, l'aggregazione dei benefici di tutti i software impiegati in questo elaborato è stata esaustiva e positiva. Attraverso l'utilizzo di risorse computazionali molto più estese, le potenzialità fornite da questa soluzione di analisi ad alta fedeltà crescono esponenzialmente.

Bibliografia

- [1] Biancolini M. E., (2017), *Fast Radial Basis Functions for Engineering Applications*.
- [2] *RBF Morph - User's Guide, 2023*.
- [3] Giorgio Graziani, (2010), *Aerodinamica*.
- [4] John D. Anderson, *Fundamentals of Aerodynamics*.
- [5] Stephen B. Pope, *Turbulent flows*.
- [6] Lopez A. Biancolini M.E. (2021), *Ottimizzazione di flussi esterni ed interni mediante metodi CFD adjoint e mesh morphing*.
- [7] Egbert Torenbeek, *Synthesis of Subsonic Airplane Design*.
- [8] *Ansys Fluent - User's Guide, 2023*.

Appendice A

Codici in linguaggio python utilizzati per il caso di studio del velivolo.

Codice 1

Creazione del DOE e del CAD relativo ad ogni DP

```
# Importazione delle librerie
import numpy as np
from scipy.stats import qmc
import pandas as pd
import os
import subprocess

# Limite inferiore e superiore per i 5 parametri scelti
bounds_l = [7,0.1,51,1,2]
bounds_up = [11,0.2,57,3,4]
n = 5 # numero di parametri
m= 50 # numero di campioni
DP = np.linspace(1,m+2,m+2,dtype = int) # array da 1 a 52

sampler = qmc.LatinHypercube(d=n, strength=1, optimization="random-cd")
# matrice campione 50 righe 5 colonne
sample = sampler.random(m)
# valori del campione riscalati tra i limiti
sample = qmc.scale(sample, bounds_l, bounds_up)
# aggiunta dei campioni con i limiti
sample = np.append(sample,bounds_l).reshape(m+1,n)
sample = np.append(sample,bounds_up).reshape(m+2,n)

datapanda = pd.DataFrame(list(zip(DP,sample[:,0],sample[:,1],sample[:,2]...
,sample[:,3],sample[:,4])), columns=['DP','wing:aspect','wing:thickr',...
'wing:xroot','wing:alphab','fuse:power'])
datapanda.to_csv('points_LHS_plane.csv', index=False)
```

```
for i in range (len(sample)):
    P1 = sample[i,0]
    P2 = sample[i,1]
    P3 = sample[i,2]
    P4 = sample[i,3]
    P5 = sample[i,4]
    os.chdir('C:\\Users\\..percorso..\\EngSketchPad\\data')
    newrow1=str('DESPMTR wing:aspect' +str(P1)+ ' # aspect ratio')
    newrow2=str('DESPMTR wing:thickr' +str(P2)+ ' # thickness ratio at root')
    newrow3=str('DESPMTR wing:xroot ' +str(P3)+ ' # xloc at root LE')
    newrow4=str('DESPMTR wing:alphab' +str(P4)+ ' # setting angle at break')
    newrow5=str('DESPMTR fuse:power ' +str(P5)+ ' # super-ellipse power of fus')
    newrow6=str('DUMP    baseline_mod'+str(i)+' .stp')
    with open('baseline_mod.csm', 'r') as file:
        lines = file.readlines()
    lines[7] = newrow1 + '\n'
    lines[14] = newrow2 + '\n'
    lines[22] = newrow3 + '\n'
    lines[16] = newrow4 + '\n'
    lines[33] = newrow5 + '\n'
    lines[237] = newrow6 + '\n'
    with open('baseline_mod.csm', 'w') as file:
        file.writelines(lines)
    os.chdir("C:\\Users\\..percorso..\\EngSketchPad\\bin")
    command = 'serveESP ..\\data\\baseline_mod -batch'
    subprocess.run(command, shell=True, check=True)
```

Codice 2

Codice che genera le cartelle di lavoro, il file mod.pts per ogni DP ed esegue l'analisi CFD sulla mesh modificata.

```

import pandas as pd
import math
import cadquery as cq
from cadquery import exporters
import os
import shutil
import glob
import subprocess
import numpy as np

file_path = 'C:\\..percorso del file..\\points_LHS_plane.csv'
df = pd.read_csv(file_path)
DOE = np.array(df)
nP = len(DOE)

old_directory = os.getcwd()

for i in range (0,nP):
    dir = str(old_directory + '\\DP'+ str(i))
    os.mkdir(dir)
    os.chdir(dir)
    shutil.copy(str(old_directory + '\\425052.cas.h5'), '425052.cas.h5')
    shutil.copy(str(old_directory + '\\425052.dat.h5'), '425052.dat.h5')
    shutil.copy(str(old_directory + '\\425052.msh'), '425052.msh')
    shutil.copy(str(old_directory + '\\baseline_mod.stp'), 'baseline_mod.stp')
    shutil.copy(str(old_directory + '\\baseline_mod' + str(i) + '.stp'),...
    str('baseline_mod' + str(i) + '.stp'))
    shutil.copy(str(old_directory + '\\roc.exe'), 'roc.exe')
    shutil.copy(str(old_directory + '\\cfg.txt'), 'cfg.txt')
    shutil.copy(str(old_directory + '\\journal.scm'), 'journal.scm')
    shutil.copy(str(old_directory + '\\duct_surf.rbf'), 'duct_surf.rbf')

# divisione per simmetria dei files .stp
x1=math.sqrt(9*4240)/2
A=df.iloc[i,1]
x2=math.sqrt(A*4240)/2
base=cq.importers.importStep('baseline_mod.stp')
ris1=base.faces(">Y").workplane(-x1).split(keepTop=True)
exporters.export(ris1, str('baseline_mod.step'))
aereo=cq.importers.importStep('baseline_mod'+str(i)+'.stp')
ris2=aereo.faces(">Y").workplane(-x2).split(keepTop=True)
exporters.export(ris2, str('baseline_mod'+str(i)+'.step'))

```

```
new_row = str('baseline_mod' +str(i)+ '.step')
with open('cfg.txt', 'r') as file:
    lines = file.readlines()
lines[3] = new_row + '\n'
with open('cfg.txt', 'w') as file:
    file.writelines(lines)

#esecuzione di roc.exe
command = 'roc -mesh cfg.txt'
subprocess.run(command, shell=False, check=False)
fileIn = "rbf_centres.txt"
with open(fileIn) as ifile:
    lines = (ifile.readlines())
npts = len(lines)
X1 = [0]*npts
Y1 = [0]*npts
Z1 = [0]*npts
dX1 = [0]*npts
dY1 = [0]*npts
dZ1 = [0]*npts

for i in range (0, npts):
    a,b,c,d,e,f =(lines[i].split())

    dX1[i] = (float(d))/1000
    dY1[i] = (float(e))/1000
    dZ1[i] = (float(f))/1000

    X1[i] = (float(a))/1000
    Y1[i] = (float(b))/1000
    Z1[i] = (float(c))/1000

    lines[i] = ".join(str(' '+str(X1[i]) + ' ' + str(Y1[i]) + ...
    ' ' + str(Z1[i]) + ' ' + str(dX1[i]) + ' ' +...
    str(dY1[i]) + ' ' + str(dZ1[i]) + '\n'))

with open("mod.pts",'w') as f:
    f.write(str(npts)+'\n')
    for u in range (1,npts) :
        f.write(str(lines[u-1]))
f.close ()

# avvio di fluent
command = r"D:\..percorso..\fluent" 3d -t4 -g -i journal.scm &'
subprocess.run(command, shell=True, check=True)
os.chdir(old_directory)
```

Codice 3

Codice che preleva i risultati delle forze di resistenza e portanza e genera il file CSV per l'ottimizzazione.

```

import pandas as pd
import numpy as np
import os

n = 5
m = 50
DP = np.linspace(1,m+2,m+2, dtype = int)

Drag = np.zeros(m+2)
Lift = np.zeros(m+2)
E = np.zeros(m+2)

file_path = 'C:\\..percorso..\\points_LHS_plane.csv'
df = pd.read_csv(file_path)
old_directory = os.getcwd()

for i in range (0,52):
    dir = str(old_directory + '\\DP'+ str(i))
    os.chdir(dir)
    with open('drag.txt', 'r') as file:
        lines = file.readlines()
        x=lines[12]
        Drag[i] = (x[58:67])

    with open('lift.txt', 'r') as file:
        lines = file.readlines()
        y=lines[12]
        Lift[i] = (y[58:67])

    # Calcolo l'efficienza aerodinamica
    E[i]=Lift[i]/Drag[i]

    os.chdir(old_directory)

datapanda = pd.DataFrame(list(zip(DP,df['wing:aspect'],...
df['wing:thickr'],df['wing:xroot'],df['wing:alphab'],...
df['fuse:power'],E)), columns = ['DP','wing:aspect',...
'wing:thickr','wing:xroot','wing:alphab','fuse:power',...
'Efficiency'])
datapanda.to_csv('results.csv', index=False)

```

Appendice B

FILE: baseline_mod.csm

```
# OPAM1/baseline (for GMGW2)
# written by John Dannenhoffer

# design paramaters (inspired by B787-800)

# wing
DESPMTR wing:area      4240 # area
DESPMTR wing:aspect   9.00 # aspect ratio
DESPMTR wing:taperi   0.48 # inboard taper ratio
DESPMTR wing:tapero   0.23 # outboard taper ratio
DESPMTR wing:sweep    32.2 # leading edge sweep
DESPMTR wing:dihedral  7.0  # dihedral
DESPMTR wing:break    0.37 # inboard/outboard
DESPMTR wing:alphar   5     # setting angle at root
DESPMTR wing:thickr   0.15 # thickness ratio at root
DESPMTR wing:camber   0.04 # camber ratio at root
DESPMTR wing:alphab   2     # setting angle at break
DESPMTR wing:thickb   0.15 # thickness ratio at break
DESPMTR wing:camberb  0.04 # camber ratio at break
DESPMTR wing:alphat  -1     # setting angle at tip
DESPMTR wing:thickt   0.08 # thickness ratio at tip
DESPMTR wing:cambert  0.04 # camber ratio at tip
DESPMTR wing:xroot    54.0 # xloc at root LE
DESPMTR wing:zroot    -5.0 # zloc at root LE

# fuselage
DESPMTR fuse:noseRad   2.5  # nose radius
DESPMTR fuse:fwdLength 28   # length of forward fuselage
DESPMTR fuse:noseHeight -2  # zloc of center of nose

DESPMTR fuse:midLength 115  # length of mid fuselage
DESPMTR fuse:midWidth  20   # width of mid fuselage
```

```

DESPMTR fuse:midHeight 20 # height of mid fuselage
DESPMTR fuse:power 3 # super-ellipse power of mid and aft fuselage

DESPMTR fuse:aftLength 44 # length of aft fuselage
DESPMTR fuse:aftWidth 1 # width of aft fuselage
DESPMTR fuse:aftHeight 3 # height of aft fuselage
DESPMTR fuse:aftCenter 8.5 # zloc of aft fuselage

# vertical tail
DESPMTR vtail:area 610 # vtail area
DESPMTR vtail:aspect 1.80 # vtail aspect ratio
DESPMTR vtail:taper 0.28 # vtail taper ratio
DESPMTR vtail:sweep 45 # vtail sweep
DESPMTR vtail:thick 0.08 # vtail thickness
DESPMTR vtail:xroot 150 # xloc of root LE
DESPMTR vtail:zroot 9 # zloc of root LE

# horizontal tail
DESPMTR htail:area 610 # htail area
DESPMTR htail:aspect 1.80 # htail aspect ratio
DESPMTR htail:taper 0.28 # htail taper ratio
DESPMTR htail:sweep 40 # htail sweep
DESPMTR htail:dihedral 12 # htail dihedral
DESPMTR htail:thick 0.08 # htail thickness
DESPMTR htail:xroot 145 # xloc of root LE
DESPMTR htail:zroot 5 # zloc of root LE

# constant parameter to tell if NACA airfoils should have sharp trailing edges
CONPMTR SHARP_TE 0 # set to 0 for standard defn, 1 for sharp TEs

# compute wing critical locations / dimensions
SET wing:yroot 0
SET wing:ytip -sqrt(wing:aspect*wing:area)/2
SET ybreak wing:ytip*wing:break
SET wing:xtip wing:xroot-wing:ytip*tand(wing:sweep)
SET xbreak wing:xroot-ybreak*tand(wing:sweep)
SET wing:ztip wing:zroot-wing:ytip*tand(wing:dihedral)
SET zbreak wing:zroot-ybreak*tand(wing:dihedral)
SET chordr wing:area/((wing:yroot-ybreak)*(wing:taperi+1)+(ybreak-...
wing:ytip)*wing:taperi*(wing:tapero+1))
SET chordb chordr*wing:taperi
SET chordt chordb*wing:tapero

# lay out left wing
MARK
UDPRIM naca camber wing:camber thickness wing:thickr sharpte SHARP_TE
SCALE chordr
ROTATEX 90 0 0
ROTATEY wing:alphar 0 0
TRANSLATE wing:xroot wing:yroot wing:zroot

```

```

UDPRIM  naca      camber wing:camberb  thickness wing:thickb  sharpTE SHARP_TE
SCALE   chordb
ROTATEX 90 0 0
ROTATEY wing:alphab 0 0
TRANSLATE xbreak      ybreak      zbreak

UDPRIM  naca      camber wing:cambert  thickness wing:thickt  sharpTE SHARP_TE
SCALE   chordt
ROTATEX 90 0 0
ROTATEY wing:alphan 0 0
TRANSLATE wing:xtip  wing:ytip  wing:ztip
RULE
  ATTRIBUTE name $leftWing

# right wing
STORE   foo
RESTORE foo
RESTORE foo
  ATTRIBUTE name $rightWing
MIRROR 0 1 0
JOIN

# compute fuselage critical locations / dimensions
SET     fuse:Length fuse:fwdLength+fuse:midLength+fuse:aftLength

SET     x1          fuse:noseRad
SET     z1          fuse:noseHeight
SET     r1          fuse:noseRad
SET     x2          fuse:fwdLength/2
SET     z2          fuse:noseHeight/2
SET     r2          (fuse:midHeight+fuse:midWidth)/8+fuse:noseRad

SET     L          hypot(x2-x1,z2-z1)

SET     x3          x2-r2/L*(z2-z1)
SET     z3          z2+r2/L*(x2-x1)
SET     x5          x2+r2/L*(z2-z1)
SET     z5          z2-r2/L*(x2-x1)

SET     D          hypot(x3-x1,z3-z1)

SET     phi        atan2(z3-z1,x3-x1)+pi(0.5)-asin(r1/D)
SET     x4          x1+r1*cos(phi)      # upper tangency
SET     z4          z1+r1*sin(phi)

SET     phi        atan2(z5-z1,x5-x1)-pi(0.5)+asin(r1/D)
SET     x6          x1+r1*cos(phi)      # lower tangency
SET     z6          z1+r1*sin(phi)

```



```

# build fuselage nose
SPHERE x1 0 z1 r1
  ATTRIBUTE name $nose
ROTATEX 90 0 z1

# trim nose to part that is exposed
SKBEG x6 -2*r1 z6 -2*r1
  LINSEG x4 -2*r1 z4 -2*r1
  LINSEG x4-2*r1 -2*r1 z4 -2*r1
  LINSEG x6 -2*r1 z6-2*r1 -2*r1
  LINSEG x6 -2*r1 z6 -2*r1
SKEND
EXTRUDE 0 4*r1 0 4*r1
INTERSECT

STORE nose

# build rest of fuselage
MARK
  # extract back of nose
  RESTORE nose
  SELECT face @nbody-2 6
  EXTRACT @iface

  # forward break
  UDPRIM supell rx r2 ry r2 n 2
  ROTATEY 90-atan2(z2-z1,x2-x1)*180/pi(1) 0 0
  TRANSLATE x2 0 z2

  # mid section
  UDPRIM supell rx fuse:midHeight/2 ry fuse:midWidth/2 n fuse:power
  ROTATEY 90 0 0
  TRANSLATE fuse:fwdLength 0 0

  UDPRIM supell rx fuse:midHeight/2 ry fuse:midWidth/2 n fuse:power
  ROTATEY 90 0 0
  TRANSLATE fuse:fwdLength+fuse:midLength 0 0

  # tail section
  UDPRIM supell rx fuse:aftHeight/2 ry fuse:aftWidth/2 n fuse:power
  ROTATEY 90 0 0
  TRANSLATE fuse:Length 0 fuse:aftCenter
RULE 1
  ATTRIBUTE name $fuselage

# join nose to rest of fuselage
RESTORE nose
JOIN 1e-5
UNION

```

```

# compute vertical tail critical locations / dimensions
SET      semispan      sqrt(vtail:area*vtail:aspect)
SET      chordr        2*vtail:area/semispan/(1+vtail:taper)
SET      chordt        chordr*vtail:taper
SET      vtail:xtip    vtail:xroot+semispan*tand(vtail:sweep)
SET      vtail:ztip    vtail:zroot+semispan

# lay out vertical tail
MARK
  UDPRIM   naca   thickness vtail:thick  sharpTE SHARP_TE
  SCALE    chordr
  TRANSLATE vtail:xroot 0  vtail:zroot

  UDPRIM   naca   thickness vtail:thick  sharpTE SHARP_TE
  SCALE    chordt
  TRANSLATE vtail:xtip 0  vtail:ztip
RULE
  ATTRIBUTE name $vtail
UNION

# compute horizontal tail critical locations / dimensions
SET      span          sqrt(htail:area*htail:aspect)
SET      chordr        2*htail:area/span/(1+htail:taper)
SET      chordt        chordr*htail:taper
SET      htail:xtip    htail:xroot+span*tand(htail:sweep)
SET      htail:ytip    -span
SET      htail:ztip    htail:zroot+span*tand(htail:dihedral)

# lay out left horizontal tail
MARK
  UDPRIM   naca   thickness htail:thick  sharpTE SHARP_TE
  ROTATEX  90    0  0
  SCALE    chordr
  TRANSLATE htail:xroot 0  htail:zroot

  UDPRIM   naca   thickness htail:thick  sharpTE SHARP_TE
  ROTATEX  90    0  0
  SCALE    chordt
  TRANSLATE htail:xtip htail:ytip htail:ztip
RULE
  ATTRIBUTE name $leftHtail

# right htail too
STORE    foo
RESTORE  foo
RESTORE  foo
  ATTRIBUTE name $riteHtail
MIRROR  0  1  0
JOIN
UNION

```

```
# adjust tessellation parameters
SELECT body @nbody
  ATTRIBUTE .tParams "1.5; 0.25; 15;"

# save .stp and .egads files
DUMP      baseline.stp
DUMP      baseline.egads

end
```

Ringraziamenti

Quest'ultima pagina è riservata a tutti coloro che mi sono stati vicini e che mi hanno supportato durante questo percorso universitario. Innanzitutto esprimo sincera gratitudine alla mia famiglia per tutti i consigli e gli aiuti offerti durante questi anni.

Un particolare ringraziamento va ai miei relatori, Federico Dalla Barba e Marco Biancolini, per la disponibilità e l'interesse dimostrato. Un grazie di cuore al mio correlatore Andrea Lopez, persona sempre cordiale e disponibile che mi ha consentito di affrontare con positività ogni tipo di ostacolo incontrato.

Un abbraccio affettuoso anche al mio nipotino Pietro per i momenti di serenità e divertimento che abbiamo trascorso assieme in questi mesi.

Infine, un grazie sincero ai miei nonni per tutto quello che hanno fatto per me. In particolare, dedico questo lavoro a mio nonno Vincenzo e a mia nonna Pina, venuta a mancare proprio alla fine di questo percorso. A loro mi sento di rivolgere un pensiero particolare per il sostegno morale che non mi hanno mai fatto mancare e per l'energia che continuano a trasmettermi ogni giorno.